NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A THREE-DIMENSIONAL TRANSONIC, POTENTIAL FLOW COMPUTER PROGRAM, ITS CONVERSION TO IBM FORTRAN AND UTILIZATION

by

Jack Paschall III

December 1983

Thesis Advisor: R. D. Zucker

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Three-Dimensional Transonic, Potential Flow Computer Program, Its Conversion to IBM Fortran and Utilization		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1983 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jack Paschall III		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE December 1983 13. NUMBER OF PAGES 152
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release, Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Program; Inviscid, Three-Dimensional Transonic Potential Flow Over Wings or Wing-Body Combinations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis describes the conversion of a computer program from Fortran IV for the NOS 1.2 operating system of the CYBER 175 or CDC 6600 computer to Fortran IV compatible with the Naval Postgraduate School IBM 3033 system. The converted program, called FLO27, calculates the inviscid, three-dimensional transonic potential flow over wings or wing-body combinations. The data input to FLO27 is

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

extensive; therefore, an interactive program was developed to aid the user in building the required input data file.

Approved for Public Release, Distribution Unlimited

A Three-Dimensional Transonic, Potential Flow Computer Program, Its Conversion to IBM Fortran and Utilization

by

Jack Paschall III
Commander, United States Navy
B.S., Oregon State University, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1983

ABSTRACT

This thesis describes the conversion of a computer program from Fortran IV for the NOS 1.2 operating system of the CYBER 175 or CDC 6600 computer to Fortran IV compatible with the Naval Postgraduate School IBM 3033 system. The converted program, called FLO27, calculates the inviscid, three-dimensional transonic potential flow over wings or wing-body combinations. The data input to FLO27 is extensive; therefore, an interactive program was developed to aid the user in building the required input data file.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

I would like to acknowledge the most valuable assistance of the W. R. Church computer center personnel in the successful completion of this thesis. In particular, the guidance of Mr Dennis Mar was instrumental in directing my initial efforts. I am grateful to Dr Robert Zucker for providing the constant attention and encouragement which kept me working at an even pace. In addition, I am deeply appreciative of the love and support provided by my wife Tricia and our two children. Most of all I thank God for the strength, determination and peace which has sustained me and brought this project to a successful completion.

I. INTRODUCTION

In the Aeronautical Engineering curriculum graduate level aerodynamics course, AE-4501, the students are exposed to two computer programs. One of these, prepared by the Douglas Aircraft Company, analyzes the potential flow around three-dimensional wings but is limited to incompressible flow [Ref. 1]. The other program, prepared by Cebeci, calculates the friction drag for two dimensional incompressible flow over airfoils [Ref. 2]. A serious defect of these programs is that they are not state-of-the-art computer programs. The Douglas program does not consider the effects of compressibility and the boundary layer program, in addition to being restricted to incompressible flow, does not predict the laminar to turbulent transition point.

A. BACKGROUND

In 1980 the Department of Aeronautics at the Naval Postgraduate School acquired an intricate computer program recently developed by the Boeing Commercial Airplane Company. This state-of-the-art program calculates three-dimensional transonic flow over wings and bodies in

both the outer-inviscid flow region governed by the transonic potential equation and the thin layer in which the first order, compressible boundary layer equations are assumed to be valid.

The Boeing program as received was designed to be executed on a CDC 6600 or a CYBER 175 computer and was written using CDC FORTRAN IV extended language. This thesis therefore was primarily concerned with the conversion of the program to FORTRAN IV extended compatible with the Naval Postgraduate School's (NPS) IBM 3033 system. The large modular program was divided so that the potential flow analysis portion could be run separately. Simplified instructions for use of the program were also prepared.

B. VISCOUS/INVISCID SYSTEM OF PROGRAMS

The Viscous/Inviscid Wing System (VIWS) of programs calculates three-dimensional transonic flow over wings and wing body combinations including details of the laminar or turbulent flow in the three-dimensional viscous boundary layer. The flow field is calculated in two overlapping regions: an outer inviscid flow region governed by the transonic potential equation, and a thin boundary layer in which the first order, three-dimensional, compressible

boundary layer equations are assumed to hold and in which the effects of surface heat and mass transfer can be computed. A list of the VIWS of programs is presented in Table I.

TABLE I

Viscous/Inviscid Wing System of Programs

Program Name	Description
FLO27	Jameson-Caughey inviscid, transonic wing code
A411IN	Reads geometry & velocity data, constructs coordinate system
VWIN	Potential flow boundary layer interface
A411AC1	Three-dimensional boundary layer program
INTERP	Boundary layer potential flow interface
A411P1 A411P2 A411PS	Graphics display programs

The basic sequence of calculations used by the VIWS to obtain matched viscous and inviscid solutions consists of an iterative loop in which the inviscid outer flow analysis and the boundary layer analysis are performed sequentially. The iterative sequence is continued until either convergence (satisfactory matching) is achieved, or the maximum number of iterations specified by the user has been performed. The VIWS programing sequence is shown schematically in Fig. 1.1.

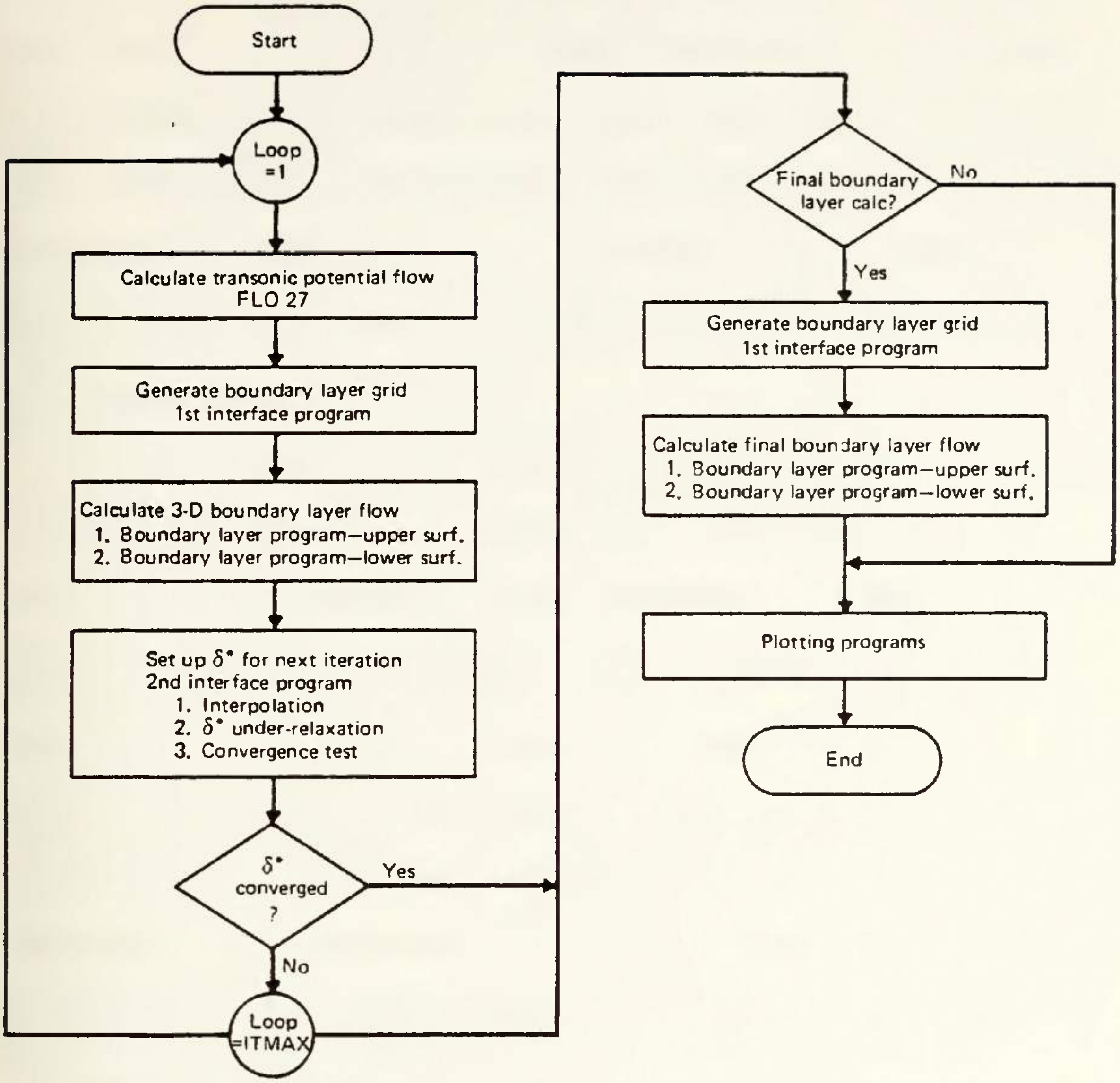

Figure 1.1. Viscous/Inviscid Interaction Procedure

The potential flow is calculated for the bare wing during the first iteration. In subsequent iterations, the effect of the boundary layer flow on the outer inviscid flow is

felt as a modification to the wing shape through the addition of the boundary layer displacement thickness. Convergence is recognized, and the iterations are stopped, when the maximum change between the new and old displacement thickness, expressed as a fraction of the maximum displacement thickness, is less than the convergence tolerance chosen by the user.

The VIWS utilizes the Jameson-Caughey transonic inviscid wing program FLO27, to carry out the potential flow analysis. The boundary layer analysis is performed by a finite difference boundary layer prediction program developed by the Boeing Commercial Airplane Company. The basic theory behind the boundary layer program is contained in [Ref. 3]. A detailed description of the VIWS of programs (excluding the potential flow program FLO27) is contained in [Ref. 4]. A basic guide to the use of the VIWS of programs is contained in [Ref. 5].

II. POTENTIAL FLOW PROGRAM FLO27

Because of the extensive length and number of program modules in the VIWS, the Potential Flow Program, FLO27, was singled out for conversion. It was anticipated that FLO27 would be run separately at first and recombined with the other program modules at some later date when these modules were themselves translated for execution on the IBM 3033 computer.

A. RE-PROGRAMING

The Potential Flow Program, hereafter called FLO27, was received on magnetic tape and loaded into the IBM 3033 mass storage system using the Job Control Language (JCL) routines presented in Appendix A. The magnetic tape contained twenty (20) total files in which the format was 9 track, 1600 CPI, unlabled. The card image format for the sixteen (16) program files is 80 characters per record and the four (4) output files contain 150 characters per record. The program and output files on the original CDC tape are listed in Table II.

The FLO27 program was converted to FORTRAN IV extended suitable for execution on the IBM computer using the NPS CDC

TABLE II

CDC Magnetic Tape Files

File/Records	Name	Description
1 /2356	FLO27	Potential Flow Program
2 /3194	A411IN	Reads geometry & velocity data constructs coordinate system
3 / 378	VWIN	Potential flow boundary-layer interface
4 /6611	A411A01	Three-dimensional boundary layer program
5 /1977	INTERP	Boundary-layer potential flow interface
6 / 688	A411PS	Streamline plots
7 / 211	A411P1	One-dimensional plots
8 / 586	A411P2	Contour plots
9 / 70	COUPLE	Procedure files
10 / 158	ITER	
11 / 7	DATAIN	
12 / 78	FINAL	
13 / 434	BOEB1	Boeing McLean computer program
14 / 36	CONTPLT	Contour plots
15 / 17	CORDPLT	One-dimensional plots
16 / 40	STREPLT	Streamwise plots
17	OUTF27	Output from FLO27
18	OUTIFC	Output from VWIN
19	OUT411L	Output from boundary-layer, lower surface
20	OUT411U	Output from boundary-layer, upper surface

to IBM conversion guide [Ref. 6]. The first step taken consisted of program compilation using the WATFIV compiler with its extended error messages. The listing which was

produced flagged all areas of the program which required revision. Program changes were accomplished utilizing this WATFIV listing. Some of the more general and repetitive changes are listed in Table III.

TABLE III

FLO27 Re-Programing Changes

CDC Code	IBM Code Change
Variables: FREAD, FREAF, FWRIT, FWRIF, IREAD, IREAF, IWRIT, IWRIF	Eliminated from program
WRITE(IWRIT,600)	WRITE(6,600)
READ(IREAF,500)	READ(5,500)
READ 7, WRITE 7 or REWIND 7	Changed to READ 14, WRITE 14 or REWIND 14
Call SECOND(T)	Step eliminated
Call SSWITCH(1,ISTOP)	Call SLITET(1,ISTOP)
Delimiter of form *	Replaced by '
Comment cards with *	Replaced by C
LEVEL statement	Step eliminated
If(UNIT(N).GT.0.) GO TO	All of this type eliminated

The most difficult change to make occurred with the CDC Buffer IN or Buffer OUT statements which were used in the program to transfer portions of a three-dimensional array into and out of main memory. The Buffer routines reduce the memory size required to execute the program. This statement type occurred in the main program and several of the subroutines.

The change required to translate this statement is presented below with the CDC code preceeding the IBM FORTRAN.

```
BUFFER OUT (N3,1) (G(1,1,1),G(MX,MY,1))  changed to
WRITE(N3) ((G(I,J,1),I=1,MX),J=1,MY)     and
BUFFER IN (N1,1) (G(1,1,M),G(MX,MY,M))   changed to
READ(N1,ERR= ) ((G(I,J,M),I=1,MX),J=1,MY)
```

The variable ERR was assigned the GO TO statement number of the UNIT statement immediately following the BUFFER IN line of code. As an example, if the UNIT statement following the BUFFER IN code was - If(UNIT(N1).GT.0.) GO TO 151, then the number 151 was assigned to variable ERR following the equal sign. All CDC UNIT statements were eliminated from the FLO27 source code per Table II.

In addition to the program changes required to run FLO27 on the IBM computer, several lines of code were added to modify the output format to a more usable form. A subroutine, VERTEC, which calls the Versatec plotter was also added to enhance program usefulness. This plotting routine is user controlled through an input variable and is ex lained in the next section. The modified FLO27 program source code is presented in Appendix E.

To facilitate program data entry several input variables which had recommended values were initialized to these values within the Main program and the subroutine GEOM. The initialized input variables and their values are presented in Table IV.

TABLE IV

Initialized Input Variables

AREA	VARIABLE NAME	INITIALIZED VALUE
MAIN Prgm.	XSCAL	0.0
	PSCAL	0.0
	FCONT	0.0
	P20	0.7
	P30	1.0
	FSMOO	0.0
	PTMAP	0.0
	BLCP	0.0
	WEIG	1.0
	PTCK	0.0
	FIX	0.0
	YSYM	0.0
Subrt. GEOM	FNB	2.0
	PX	0.0
	PZ	0.0
	TRL	0.0
	SLT	0.0
	XSING	0.0
	YSING	0.0

A complete description of each input variable in Table IV can be found on pages 19 through 23 of [Ref. 5].

B. PROGRAM DESCRIPTION

The FLC27 program is a computer code written to analyze the transonic flow over a wing alone or a wing on a cylindrical fuselage. It uses a finite-volume formulation to solve the exact potential flow equation in conservative form. In the development of the equations, the basic assumptions are; steady flow, no heat or work transfer, isentropic flow, irrotational flow, no body forces and a perfect gas. The velocity vector in cartesian coordinates is

$$\vec{V} = u\hat{i} + v\hat{j} + w\hat{k} \tag{2.1}$$

where u, v and w are the velocity components. The continuity equation, assuming steady flow, is

$$\frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) + \frac{\partial}{\partial z}(\rho w) = 0 \tag{2.2}$$

Next a velocity potential is introduced such that the velocity components are calculated as the gradient of this potential.

$$u = \phi_x \ , \ v = \phi_y \ , \ w = \phi_z \tag{2.3}$$

With the introduction of the velocity potential, the continuity equation 2.2 becomes

$$\frac{\partial}{\partial x}(\rho\phi_x) + \frac{\partial}{\partial y}(\rho\phi_y) + \frac{\partial}{\partial z}(\rho\phi_z) = 0 \qquad (2.4)$$

Assuming no heat or work transfer, the energy equation can be written as

$$T\left[1 + \frac{(\gamma - 1)}{2} M^2\right] = T_\infty\left[1 + \frac{(\gamma - 1)}{2} M_\infty^2\right] \qquad (2.5)$$

The flow is assumed to be uniform in the far field. On the surface of the body, the normal velocity component is zero. The velocities and densities of the near field are normalized using the free stream velocity and density, thus $V_\infty = 1$ and $\rho_\infty = 1$. Using the assumptions that the flow is isentropic and a perfect gas, the energy equation 2.5 can be shown to be

$$\rho = \left[1 + \frac{(\gamma - 1)}{2} M_\infty^2 \left(1 - V^2\right)\right]^{\frac{1}{\gamma - 1}} \qquad (2.6)$$

With equations 2.5 and 2.6 there are only two unknowns, ϕ and ρ. They can be solved, subject to the boundary condition of flow tangency, using a finite volume technique. The basic numerical scheme for the solution is the

construction of a mesh from small volume elements (cubes) which are packed around the wing or wing-body configuration. The cubes in the computational domain are separately mapped to distorted cubes in the physical domain by independent transformations from local coordinates X, Y and Z to Cartesian coordinates x, y and z. The mesh points are the vertices (corners) of the mapped cubes. The velocity potential and density are calculated at each vertex in the mesh. The pressure distribution can then be calculated from

$$P = \frac{\rho^{\gamma}}{\gamma M_{\infty}^{2}} \tag{2.7}$$

In the event that the local flow velocity becomes supersonic and shocks occur, these are handled in the usual manner by insuring that:

1) The tangential velocity components are equal on each side of the shock.

2) Continuity is maintained by keeping the product of ρU_n constant across the shock (where U_n is the normal velocity component).

3) Discontinuous expansions (corresponding to an "expansion shock") are excluded from the flow field.

The assumption of isentropic flow along with the existence of shocks presents a contradiction which can only be resolved by limiting the flow to very weak shocks for which entropy and vorticity generation may be ignored. Thus, solutions will be valid only for subsonic free stream velocities.

The main three-dimensional array containing the potential function data is stored on disk, and special unformated input/output statements are used to bring planes of data into central computer memory and to store updated planes of data back on the disk. In the construction of the computational coordinate system, a Joukowski transformation is used to transform the cylindrical fuselage to a vertical slit and then a sheared parabolic transformation is used in planes containing the airfoil sections. A detailed mathematical formulation of the potential flow analysis is contained in [Ref. 7].

1. Program Input

The input to FLO27 consists of variables which are read with an 8F10.6 FORMAT. Each input card has a title card which precedes it. This title card contains the input variable name and effectively labels the input data for easy

reference. The title for each input data is placed in the same column as the input data it labels. The title cards are read with a 20A4 FORMAT. All numerical input values are real numbers. The following data deck, listed card by card, is the minimum input data required for a simple wing analysis. Each "card" can be interpreted as one line of data on your terminal. A complete sample data set is presented in Appendix C.

CARD 1 The Run Title (64 characters maximum)

CARD 2 Title card for the input variables

FNX, FNY, FNZ, FMESH and FPLOT

CARD 3

Cols. 1-10 FNX - Number of computational cells in the chordwise direction for the initial mesh.

MAX = 160/2**n, where n = FMESH -1. (See Cols. 31-40 for FMESH)

Cols. 11-20 FNY - Number of computational cells in the normal direction from the airfoil surface for the initial mesh.

MAX = 16/2**n, where n = FMESH -1.

Cols. 21-30 FNZ - Number of computational cells in the spanwise direction for the initial mesh.

MAX = 32/2**n, where n = FMESH -1.

Cols. 31-40 FMESH - Determines the number of times a program generated computational mesh is refined. Enter only 1.0, 2.0 or 3.0 for coarse, medium or fine mesh. If 3.0 is selected the program will calculate flow over the wing for the coarse mesh then half the mesh size (medium), recalculate, then half the mesh again (fine) and do a final potential flow calculation. Output parameters are printed for each mesh size for which calculations were performed.

Cols. 41-50 FPLOT - Output flag

0.0 = Normal output without printer-plot of Cp

1.0 = Normal output with printer-plot of Cp at each computational mesh point for each wing section.

2.0 = Normal output with Versatec plots of Cp versus X/C for each wing section of the final mesh.

CARD 4 Title card for the input variables FIT, COVO and P10

CARD 5-M One card for each computational mesh. Total number of cards equal to M = FMESH.

Cols. 1-10 FIT - A parameter which fixes the maximum number of iterations the program will use to converge the velccity pctential to a specified tolerance (COVO). This parameter must be repeated for each mesh refinement.

Cols. 11-20 COVO - Velocity potential convergence criteria. This input variable is also entered for each selected mesh. A value of 0.000001 is reccmmended.

Cols. 21-30 P10 - This parameter determines the subsonic point relaxation factor for the specified mesh size. A value of less than 2.0 must be entered for each designated mesh. Recommended values are: 1.6 fcr coarse, 1.3 for medium and 1.2 for the fine mesh.

CARD 6 Title card for the input variables FMACH, YA, AL and CDO

CARD 7

Cols. 1-10 FMACH - Free stream Mach number

Cols. 11-20 YA - Yaw angle in degrees

Cols. 21-30 AL - Angle of attack in degrees

Cols. 31-40 CDO - Drag coefficient due to skin friction. Unless known, an estimated value of 0.01 is recommended.

CARD 8 Title card for the input variables ZSYM, FNS, SWEEP, DIHED and FUS

CARD 9

Cols. 1-10 ZSYM - The wing planform symmetry trigger.

0.0 = Yawed wing, has no spanwise symmetry

1.0 = Swept wing, has spanwise symmetry

Cols. 11-20 FNS - This input variable tells the program the total number of wing sections you have selected to define the wing half span. The number must be at least three (3) but not more than eleven (11) sections.

Cols. 21-30 SWEEP - Leading edge sweep angle in degrees.

Cols. 31-40 DIHED - Dihedral angle in degrees. See Fig. 2.1.

Cols. 41-50 FUS - Input the fuselage radius. Enter 0.0 for a wing-alone case.

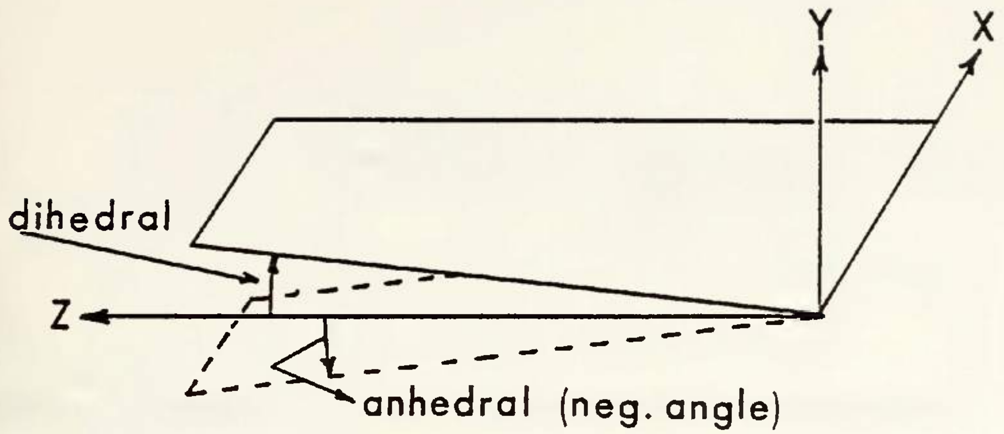

Figure 2.1. Dihedral Angle

Data input cards from 10 through 15 are used for defining wing planforms and section geometrics. For the first wing section, all data cards from card 10 through card 15 must be used. For the second and subsequent sections there is an option for skipping the wing section defining data (cards 12 through 15) and copying the data from that of the previous section. This option is controlled by the input variable FSEC. If this option is not used, data cards from 10 through 15 must be repeated for each wing defining section. The number of wing sections which are defined is input with the variable FNS. Remember, up to 11 sections may be defined, and a minimum of 3 sections must be defined. All wing planform and section defining geometrics must be in consistent units. Wing planform and section defining quantities are presented in Fig. 2.2.

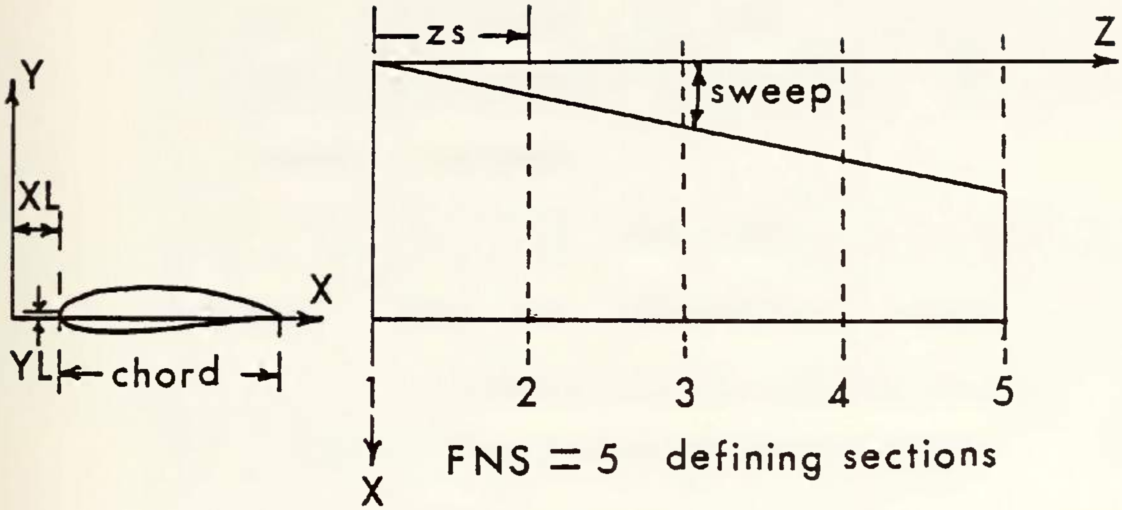

Figure 2.2. Wing Defining Geometry

CARD 10 Title card for the input variables

ZS, XL, YL, CHORD, THICK, AT and FSEC

CARD 11

Cols. 1-10 ZS - The section spanwise coordinate (Start at the centerline and work outboard)

Cols. 11-20 XL - Section leading edge X coordinate

Cols. 21-30 YL - Section leading edge Y coordinate

Cols. 31-40 CHORD - Section chord length

Cols. 41-50 THICK - The thickness scaling factor can be used to scale all Y coordinates of the wing section. Thus percent thickness and camber are increased (or decreased) accordingly. Use 1.0 if no scaling is desired.

Cols. 51-60 AT - The twist angle of each section (geometric twist) measured from the X axis to the chord line. A positive twist angle reduces the section angle of attack and gives "washout". Use 0.0 for no twist.

Cols. 61-70 FSEC - This is a flag which determines whether or not the program reads wing section defining geometry from a previous wing section or from new defining geometry. For the first section defined you <u>must</u> set FSEC to 1.0. Following the first section, if you define new section geometry then use FSEC = 1.0. If you want the program to read the section geometry defined from the previous section then set FSEC = 0.0.

CARD 12 Title card for the input variable FN

CARD 13

Cols. 1-10 FN - This variable contains the number of points which define the upper and lower surface of the section. A maximum of 161 points may be used.

CARD 14 Title card for the input variables XP(I) and YP(I)

CARDS 15-1 to 15-N Total number of cards equals N, where N = integer part of (FN+2)/3.

The X and Y coordinates at each point are entered in pairs, three points to a card. (See Appendix C for sample input)

Cols. 1-10 XP(I) - X coordinate of the wing section point

11-20 YP(I) - Y coordinate of the wing section point

21-30 defining X coordinate for next point

31-40 defining Y coordinate for next point

41-50 defining X coordinate for following point

51-60 defining Y coordinate for following point

The X and Y coordinates of the wing section defining points must be entered starting with the upper surface trailing edge point and proceeding along the upper surface to the leading edge, and returning along the lower surface to the lower surface trailing edge point. It is very important to define the section leading edge with a large number of closely spaced points. Suggest at least 0.05 spacing or less between X coordinate values from 0.1 X/C to the leading edge, X/C = 0.0. Each X and Y coordinate point is normalized using the chord length for that section. Section defining geometrics are illustrated in Fig. 2.3.

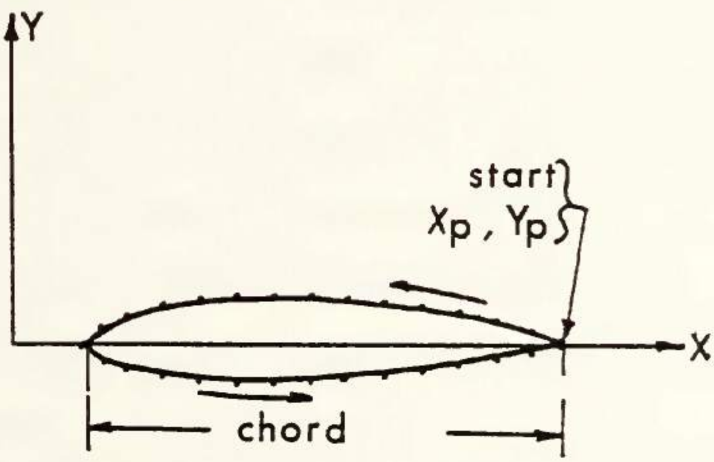

Figure 2.3. Section Defining Geometry

CARD 16 Title card containing the words in Cols. 1-80

END OF CALCULATIONS

CARD 17 Title card for the input variable

FNX

CARD 18

Cols. 1-10 FNX - This variable indicates the end of a set of calculations and must be set equal to 0.0. Its purpose is to indicate that the program has run to completion.

2. Program Output

Output from the FLO27 program varies with the value of the input variable FPLOT. When FPLOT is set equal to 0.0 a normal output is produced. This normal output contains (in order of occurance): refined input geometry data including trailing edge slope and angle calculations; iterative solution of the potential flow mesh; section characteristics and wing characteristics. The iterative solution, section and wing characteristic data are repeated for each mesh refinement requested. Thus, if the input variable FMESH is set equal to 3, these data are calculated and output three times. The last data in the normal output consists of the non-dimensionalized chord (X/C) and pressure

coefficient (Cp) data at each computational mesh point for each wing section calculated during the final mesh. A sample of the normal output data is presented in Appendix D and represents the output data from Appendix C input data.

If variable FPLOT is set equal to 1.0, the output data is increased considerably. This output contains the normal output plus a line printer-plot of the pressure coefficient at each computational mesh point for each wing section. The line printer-plot is produced for each wing section of each mesh refinement. The length of the output data with FPLOT set equal to 1.0 can approach 6000 records depending on the number of mesh refinements requested. These plots are of questionable value and, therefore, an alternate plotting program was developed.

When the variable FPLOT is set equal to 2.0, the normal output data is produced plus a Versatec plotting subroutine (VERTEC) is called. The subroutine outputs, via the Versatec plotter, plots of Cp versus X/C for each wing section of the final mesh calculations. This routine is simply putting into plot form the Cp and X/C numerical data contained in the normal output. A sample of the Versatec plot is presented in Fig. 2.4.

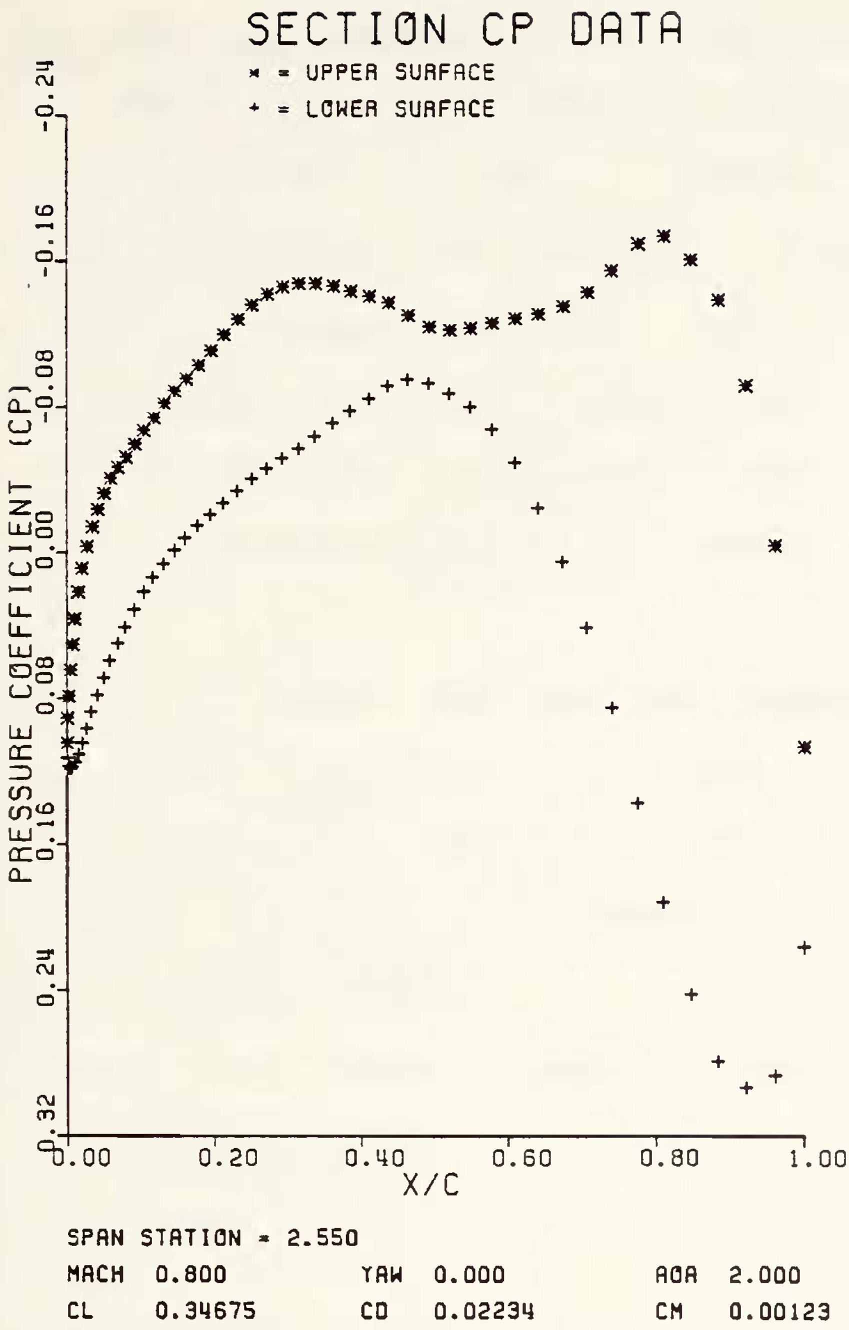

Figure 2.4. Versatec Plot of Cp vs. X/C

III. INTERACTIVE INPUT PROGRAM FLO27IN

The input data file required for the FLO27 program is extensive. Errors in input data FORMAT will cause program errors at execution time. In order to eliminate these errors and reduce the input data workload, a computer terminal interactive program was written. This interactive program, called FLO27IN, is a user-friendly way of creating an input data file for the potential flow wing analysis program FLO27. The FLO27IN program source code is presented in Appendix F.

The interactive program, FLO27IN, when executed ask's questions of the user in order to construct and write to the user's "A" disk the required FLO27 input data file. The following presents the step-by-step procedure for executing the interactive program FLO27IN.

STEP #1---Log on to any IBM 3033 interactive terminal with your user number and password.

STEP #2---Once logged on and in the CMS operation mode type:

CP LINK 0247P 191 120 RR then hit ENTER

STEP #3---The word PASSWORD will appear, Type and ENTER

AERO

STEP #4---Type and ENTER

ACC 120 D

STEP #5---Type and ENTER

LOAD FLO27IN (START

The screen will display the header for the interactive program. Answer each question presented. At the end of each question in parenthesis is the input data variable associated with that question and whether the input parameter is a real number (R) or an integer (I). Example: ==> Enter the free stream Mach number (FMACH): (R). FMACH is the input data variable for the question. As you proceed through the FLO27IN program, opportunities to review and change input data will be presented. Should it become necessary to change your input data after completing the FLO27IN program, you can simply XEDIT the created data file.

The FLO27IN program also incorporates a library which contains the wing-section defining data for a number of current wing shapes. A copy of this library is presented in Appendix B. This feature will be displayed during program execution by the use of a menu from which the user can select a pre-defined wing section or define his own.

Upon completion of user inputs to the interactive program three additional data lines are automatically written to the bottom of the input file. They are:

END OF CALCULATION

FNX

0.0

In addition, Job Control Language (JCL) cards are written to the top and bottom of the file. All JCL cards start with a // format. After FLO27IN has run to completion type and enter RELEASE 191 to release the aero disk which was linked while executing the FLO27IN program. The created data file is written to the user's "A" disk with <filename> <filetype> of FLO27 DATAIN. Additional changes can be made simply by entering the XEDIT mode and editing the file.

IV. FLO27 BATCH SYSTEM EXECUTION

The potential flow program FLO27 can be executed after the input data file has been created. The batch processor is required for FLO27 execution because of the extensive CPU time needed to run the program. While in the XEDIT mode, a standard JOB card must be added to the top of the FLO27 DATAIN file prior to submission for job execution. The JOB card has the form:

//jobname JOB (nnnn,pppp),'ident',CLASS=J

jobname = may contain up to 8 alphanumeric characters, the first of which must be alphabetic.

nnnn = your user number

pppp = project number, assigned by professor

'ident' = contains the user's own identification information. A maximum of 20 characters may be contained within the single quotation marks.

After adding the JOB card to your data file, you are ready to execute the program. Type SUBMIT FLO27 DATAIN and press ENTER. Batch runs are normally not worth waiting for. To inquire about the status of the job, enter INQ and the job name used on the JOB card or "logoff". If the system is busy and the maximum mesh size was selected, it may be several hours before your job is run.

When the job is run the output will be spooled to the batch printer located next to the VM printer in the main computer building. The title at the top of the printout for batch jobs is the name entered on the JOB card. If it is desired to have the program output data spooled directly to the terminal, it will be necessary to add one additional JCL card to the input data set. This card must be placed immediately following the JOB card and has the form:

```
//*MAIN ORG=NPGVM1.nnnnP
```

where nnnn = your user number

Inserting this card in the input data will cause all program output to be spooled to the user's virtual reader where it may be looked at, printed or transferred to his "A" disk. To enquire as to whether information is in the reader simply type RDR and hit enter, then follow the instructions on the screen.

V. PROGRAM TEST RESULTS

The FLO27 program was tested in three stages; (1) during the reprograming phase for conversion completeness, (2) after sucessful conversion with suitable wing data for program accuracy and (3) during an AE-4501 class project.

A. ACCEPTANCE TEST DATA

To test and ensure that the FLO27 program was converted to IBM compatible Fortran without error, an acceptance test data set was used. The acceptance test input and output data was supplied with the original CDC program source code. After conversion of the FLO27 program to Fortran suitable for the NPS IBM system, the acceptance test input data were run and the output results compared to the output generated by the CDC system.

Both output data sets were numerically exact when the FLO27 program was run in double precision on the IBM system. If the program was run in single precision, the numerical output values were exact to the third decimal place. The difference in single precision accuracy occurs because the CDC system uses a 64 bit word length while the IBM system word length in single precision is only 32 bits. It was

decided that the IBM single precision accuracy was satisfactory.

B. COMPARISON WITH OTHER PROGRAMS

The FLO27 program was also tested for accuracy by using the wing planform and section data from a NACA 572 wing. The data were run on both the FLO27 program and the Douglas potential flow program [Ref. 1]. The data generated by both programs was compared to wind tunnel data for the NACA 572 wing [Ref. 8]. The results are presented in Fig. 5.1 as plots of lift coefficient versus angle-of-attack. The results show that for the NACA 572 wing the FLO27 program more accurately predicts the wing lift coefficient than does the Douglas program.

C. AE-4501 CLASS PROJECT

The final test phase was conducted by introducing the FLO27 program into the AE-4501 course as a class project. This was accomplished to determine student problems/comments concerning the data input program FLO27IN and to test an additional wing shape. The wing chosen for study was that of the A-7 airplane. The A-7 wing has a distinct leading edge notch at the approximate mid-span. When the planform geometry was run with the notch included the FLO27 program

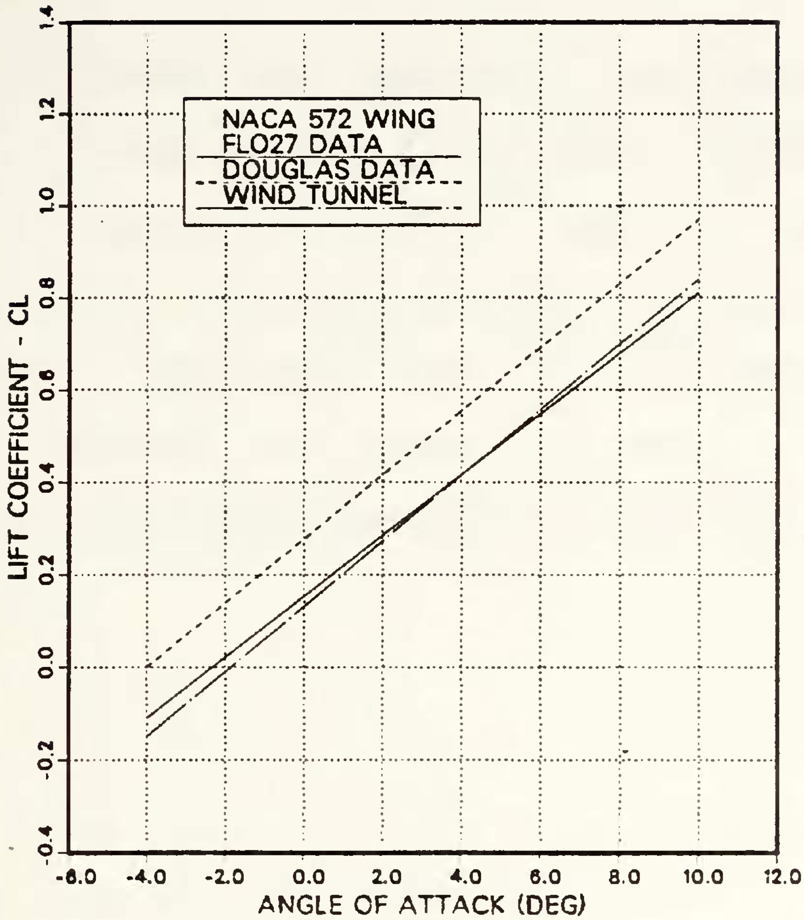

Figure 5.1. Program Calculated and Wind Tunnel Data

ran to completion but gave negative values for section and total induced drag coefficient. The value for the lift coefficient was low for the freestream Mach and angle-of-attack used. It was found that if the notch was

excluded from the wing geometry input data the program results were satisfactory both for induced drag and lift coefficient.

From the AE-4501 class experience it was determined that sharp wing planform discontinuities cannot be handled by the program. If however, the changes in shape are gradual, such as a wing glove, the program output appears to be satisfactory. Such was the case with the acceptance test case data where the wing geometry was that of the F-8 supercritical wing which incorporates a wing glove.

APPENDIX A

```
C This JCL routine allocates sufficient space on the mass
C storage system to store the entire tape contents
//JACK JOB (3266,0178),'PASCHALL-2759',CLASS=A
//*MAIN ORG=NPGVM1.3266P
//   EXEC PGM=IEFBR14
//DD1 DD UNIT=3330V,MSVGP=PUB4C,DISP=(NEW,CATLG),
//   DSN=MSS.S3266.WFLOW.DATA,SPACE=(CYL,(16,4,2))
/*
//
```

```
C This JCL routine is used to transfer all tape files to
C a partitioned data set in the mass storage system
//JACK JOB (3266,0178),'PASCHALL-2759',CLASS=J
//*MAIN ORG=NPGVM1.3266P
//COPY   PROC FILE=,MEM=
//   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD UNIT=3400-6,VOL=SER=WFLOW,DISP=(OLD,PASS),
//   LABEL=(&FILE,BLP,,IN),
//   DCB=(RECFM=F,BLKSIZE=80,DEN=3,OPTCD=Q)
//SYSUT2 DD DISP=(OLD,KEEP),
//   DSN=MSS.S3266.WFLOW.SOURCE(&MEM),
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400)
//    PEND
//    EXEC COPY,FILE=1,MEM=FLO27
//    EXEC COPY,FILE=2,MEM=A411IN
//    EXEC COPY,FILE=3,MEM=VWIN
//    EXEC COPY,FILE=4,MEM=A411A01
//    EXEC COPY,FILE=5,MEM=INTERP
//    EXEC COPY,FILE=6,MEM=A411PS
//    EXEC COPY,FILE=7,MEM=A411P1
//    EXEC COPY,FILE=8,MEM=A411P2
//    EXEC COPY,FILE=9,MEM=COUPLE
//    EXEC COPY,FILE=10,MEM=ITER
//    EXEC COPY,FILE=11,MEM=DATAIN
//    EXEC COPY,FILE=12,MEM=FINAL
//    EXEC COPY,FILE=13,MEM=BOEB1
//    EXEC COPY,FILE=14,MEM=CONTPLT
//    EXEC COPY,FILE=15,MEM=CORDPLT
//    EXEC COPY,FILE=16,MEM=STREPLT
//COPY2 PROC FILE=,MEM=,LRECL=80,BLK=6400
//   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD UNIT=3400-6,VOL=SER=WFLOW,DISP=(OLD,PASS),
//   LABEL=(&FILE,BLP,,IN),
//   DCB=(RECFM=F,BLKSIZE=&LRECL,DEN=3,OPTCD=Q)
//SYSUT2 DD DISP=(OLD,KEEP),DSN=MSS.S3266.WFLOW.DATA(&MEM),
//   DCB=(RECFM=FB,LRECL=&LRECL,BLKSIZE=&BLK)
//    PEND
//    EXEC COPY2,FILE=17,LRECL=150,BLK=6000,MEM=OUTF27
//    EXEC COPY2,FILE=18,LRECL=150,BLK=6000,MEM=OUTIPC
//    EXEC COPY2,FILE=19,LRECL=150,BLK=6000,MEM=OUT411L
//    EXEC COPY2,FILE=20,LRECL=150,BLK=6000,MEM=OUT411U
/*
//
```

```
C This JCL routine moves all source code files from mass
C storage to the MVS 004 disk which can be accessed by
C entering GET MVS then following the screen instructions
C to move source files to your disk. If you want to move
C the data files to MVS 004 then change the word SOURCE
C to DATA in the JCL program below.
//JACK JOB (3266,0178),'PASCHALL-2759',CLASS=A
//*MAIN ORG=NPGVM1.3266P
//   EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//FROM DD DISP=SHR,DSN=MSS.S3266.WFLOW.SOURCE
//INTO DD UNIT=3350,VOL=SER=MVS004,DISP=(NEW,KEEP),
//       SPACE=(CYL,(16,4,10),RLSE),DSN=S3266.SOURCE
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSIN DD *
  COPY OUTDD=INTO,INDD=FROM
/*
//
```

APPENDIX B

LIBRARY OF AIRFOIL SECTION GEOMETRIES

0 = user input section coordinate data

1 = flat plate data

2 = symmetrical wing (11% thickness at 30% chord)

3 = supercritical wing (cambered, 12% thickness at 32% chord)

4 = NACA 24-30-0 (cambered, 12% thickness at 30% chord)

5 = F-14 wing (cambered, 9.5% thickness at 37% chord)

6 = A-7 wing (7 deg droop at 20% chord, 7% thickness at 43% chord)

7 = LISSAMAN 7769 Airfoil (cambered, 11% thickness at 30% chord)

8 = NACA 0010 (symmetrical, 10% thickness at 30% chord)

9 = NACA 0010-34 (symmetrical, 10% thickness at 40% chord)

10 = NACA 0010-35 (symmetrical, 10% thickness at 50% chord)

11 = NACA 0010-64 (symmetrical, 10% thickness at 40% chord)

12 = NACA 0010-66 (symmetrical, 10% thickness at 60% chord)

13 = NACA 16-009 (symmetrical, 9% thickness at 50% chord)

14 = NACA 63-010 (symmetrical, 10% thickness at 35% chord)

15 = NACA 63A010 (symmetrical, 10% thickness at 35% chord)

16 = NACA 64-010 (symmetrical, 10% thickness at 40% chord)

17 = NACA 64A010 (symmetrical, 10% thickness at 40% chord)

18 = NACA 65-010 (symmetrical, 10% thickness at 40% chord)

19 = NACA 65A010 (symmetrical, 10% thickness at 40% chord)

20 = NACA 66-010 (symmetrical, 10% thickness at 45% chord)

APPENDIX C

THIS APPENDIX PRESENTS A COMPLETE INPUT DATA SET INCLUDING THE JCL
CARDS REQUIRED TO EXECUTE THE PROGRAM FLO27

```
// (STANDARD JOB CARD - SEE MVS USER'S GUIDE NO. MVS-01)
//   EXEC FLO27
//GO.SYSIN  DD  *
 SAMPLE DATA (NACA 572 WING SECTION)
FNX        FNY        FNZ        FMESH      FPLOT
40.0        4.C        8.0       3.0        0.0
FIT        COVC       P10
100.C      .CCC001    1.6
 20.C      .CCC001    1.3
  5.C      .0CC001    1.2
FMACH      YA         AL         CD0
0.10       0.C        2.0        0.010000
ZSYM       FNS        SWEEP      DIHED      FUS
1.0         3.C       32.3275     0.0        0.0
ZS         XL         YL         CHORD      THICK      AT         FSEC
  0.C        C.0        0.0        8.6000     1.0000     0.0       1.0
FN
 41.C
XP(I)      YP(I)
 1.0000000 0.0013000 C.9500000 0.0120000 C.9CC000C 0.0204000
 0.8CC0000 0.C376000 C.7000000 0.0515000 0.6000000 0.0642000
 0.5C00000 0.C730C00 C.4C000C0 0.0780000 C.3CC000C 0.0788000
 0.2500000 0.C768000 C.2000000 0.0726C00 0.15C000C 0.0661000
 0.1C0000C 0.C563000 C.0500000 0.0414000 C.04C000C 0.0350000
 0.0300000 C.C250C00 C.020CCC0 0.0220000 C.01C000C 0.0128000
 0.0C50000 0.C070000 0.0025000 0.0050000 0.0       0.0
 0.0C2500C-0.CC50C00 0.005000C-0.C07000C C.01C0000-0.0120000
 0.0200000-0.C168C00 C.0300000-0.0227000 C.04C000C-0.0268000
 0.05000C0-0.C302000 0.1000000-0.0375C00 C.150000C-0.0420000
 0.2000000-0.0422C00 C.250CCC0-0.0422000 C.3CC000C-0.0412000
 0.4C0000C-0.C358000 C.5000000-0.0320000 C.6CC000C-0.0260000
 0.7CC0000-0.C215C00 C.8000000-0.0150000 C.90C0000-0.0082000
 0.95000CC-0.CC48000 1.0000000-0.0013000
ZS         XL         YL         CHORD      THICK      AT         FSEC
  9.6875     6.1307     0.C        6.4500     1.C000     0.0       0.0
ZS         XL         YL         CHORD      THICK      AT         FSEC
 19.3750    12.2614     0.0        4.3000     1.0000     0.0       0.0
 END OF CALCULATION
FNX
0.0
/*
//
```

APPENDIX D

```
THIS APPENDIX PRESENTS THE FLO27 OUTPUT DATA PRODUCED FROM
THE INPUT DATA OF THE PREVIOUS APPENDIX.

 A46C MODIFIED FROM FLO27 OF ANTONY JAMESON,COURANT INSTITUTE
 THREE DIMENSIONAL WING ANALYSIS IN TRANSONIC FLOW USING FINITE VOLUME SCHEME
  NACA 572 WING SECTION
  FUSELAGE RAD
      0.0
      SWEEP             DIHED
     32.3275            0.0

 PROFILE AT Z =      0.0
      TE ANGLE         TE SLOPE          X SING           Y SING
      16.0833         -0.0720           0.0024          -0.0000


 NL =  21,  XP(NL) =     0.0

 (XP,YP)

      1.00000  -0.00130       0.95000  -0.00480       0.90000  -0.00820
      0.60000  -0.02600       0.50000  -0.03200       0.40000  -0.03580
      0.20000  -0.04220       0.15000  -0.04200       0.10000  -0.03750
      0.03000  -0.02270       0.02000  -0.01680       0.01000  -0.01200
      0.0       0.0           0.00250   0.00500       0.00500   0.00700
      0.03000   0.02900       0.04000   0.03500       0.05000   0.04140
      0.20000   0.07260       0.25000   0.07680       0.30000   0.07880
      0.60000   0.06420       0.70000   0.05150       0.80000   0.03760
      1.00000   0.00130
 SECTION DEFINITION AT Z =    0.0
        XLE             YLE                CHORD    THICKNESS RATIO      TWIST
        0.0             0.0                8.6000       1.0000           0.0
        YMIN            JMIN               YMAX         JMAX             YDIF
     -.4220E-01          10              .7880E-01       33             .1210
 SECTION DEFINITION AT Z =    9.68750
        XLE             YLE                CHORD    THICKNESS RATIO      TWIST
       6.1307           0.0                6.4500       1.0000           0.0
        YMIN            JMIN               YMAX         JMAX             YDIF
     -.4220E-01          10              .7880E-01       33             .1210
 SECTION DEFINITION AT Z =   19.37500
        XLE             YLE                CHORD    THICKNESS RATIO      TWIST
      12.2614           0.0                4.3000       1.0000           0.0
        YMIN            JMIN               YMAX         JMAX             YDIF
     -.4220E-01          10              .7880E-01       33             .1210
```

ITERATIVE SOLUTION

MACH NO	YAW		ANG OF ATTACK
0.10000	.0		2.0000
NX		NY	NZ
40		4	8
RELAX FCT 1	RELAX FCT 2		RELAX FCT 3
1.60000	.70000		1.0000

ITERATION	MAX CORRECN	I	J	K	AVG CORRECN	MAX RESIDAL	I	J	K
1	0.64577E-02	31	5	4	0.32068E-03	0.82476E-03	32	5	3
2	-0.38921E-02	21	4	4	0.40941E-03	0.16253E-02	40	4	4
3	-0.23362E-02	9	5	4	0.42671E-03	0.28477E-02	40	4	3
4	-0.21193E-02	21	4	3	0.32690E-03	-0.24366E-02	2	4	3
5	-0.13561E-02	9	5	4	0.27852E-03	-0.24832E-02	2	4	3
6	-0.12571E-02	21	4	3	0.20405E-03	-0.24482E-02	2	4	3
7	-0.84897E-03	9	5	5	0.17573E-03	-0.19958E-02	2	4	3
8	-0.74854E-03	23	4	3	0.13073E-03	-0.17460E-02	2	4	3
9	-0.52933E-03	9	5	5	0.11389E-03	-0.16529E-02	2	3	3
10	-0.43204E-03	23	4	3	0.88139E-04	-0.12835E-02	2	3	3
11	-0.32601E-03	21	3	3	0.76687E-04	0.12385E-02	40	2	3
12	0.26093E-03	33	5	5	0.61975E-04	0.14402E-02	40	2	3
13	-0.23663E-03	21	3	3	0.53340E-04	-0.16725E-02	2	2	3
14	0.17039E-03	22	3	3	0.44377E-04	-0.17377E-02	2	2	3
15	-0.16200E-03	21	3	3	0.38205E-04	-0.18561E-02	2	2	3
16	0.11217E-03	22	3	3	0.32361E-04	-0.18312E-02	2	2	3
17	-0.10598E-03	20	3	3	0.27590E-04	-0.18441E-02	2	2	3
18	0.72128E-04	39	4	3	0.23969E-04	-0.17890E-02	2	2	3
19	-0.66828E-04	19	3	3	0.21097E-04	-0.17426E-02	2	2	3
20	-0.56016E-04	8	5	3	0.18090E-04	-0.16707E-02	2	2	3
21	-0.49252E-04	9	5	3	0.16132E-04	-0.16007E-02	2	2	3
22	-0.42871E-04	8	5	3	0.13933E-04	-0.15204E-02	2	2	3
23	-0.37882E-04	9	5	3	0.12510E-04	-0.14424E-02	2	2	3
24	-0.32161E-04	8	5	3	0.10915E-04	-0.13610E-02	2	2	3
25	-0.28624E-04	9	5	3	0.98405E-05	-0.12837E-02	2	2	3
26	-0.23999E-04	8	5	3	0.87049E-05	-0.12052E-02	2	2	3
27	-0.21536E-04	9	5	3	0.78556E-05	-0.11320E-02	2	2	3
28	-0.17870E-04	8	5	3	0.70402E-05	-0.10554E-02	2	2	3
29	-0.16175E-04	9	5	3	0.63569E-05	-0.99177E-03	2	2	3
30	-0.13433E-04	8	5	3	0.57511E-05	-0.92579E-03	2	2	3
31	-0.12296E-04	9	5	3	0.52025E-05	-0.86495E-03	2	2	3
32	-0.10279E-04	4	3	3	0.47274E-05	-0.80540E-03	2	2	3
33	-0.94565E-05	9	5	3	0.42972E-05	-0.75145E-03	2	2	3
34	-0.85878E-05	4	3	3	0.39299E-05	-0.69891E-03	2	2	3
35	-0.78837E-05	4	3	3	0.35738E-05	-0.65088E-03	2	2	3
36	-0.72159E-05	4	3	3	0.32773E-05	-0.60493E-03	2	2	3
37	-0.66577E-05	4	2	5	0.29882E-05	-0.56222E-03	2	2	3
38	-0.61694E-05	4	2	5	0.27457E-05	-0.52280E-03	2	2	3
39	-0.56982E-05	4	2	5	0.25094E-05	-0.48517E-03	2	2	3
40	-0.52772E-05	4	2	5	0.23090E-05	-0.45083E-03	2	2	3

41	-0.48780E-05	4	2	5	0.21154E-05	-0.41835E-03	2	2	3
42	-0.45226E-05	3	2	5	0.19479E-05	-0.38798E-03	2	2	3
43	-0.41823E-05	3	2	5	0.17892E-05	-0.36010E-03	2	2	3
44	-0.38741E-05	3	2	5	0.16486E-05	-0.33429E-03	2	2	3
45	-0.35907E-05	3	2	5	0.15171E-05	-0.30978E-03	2	2	3
46	-0.33153E-05	3	2	5	0.13986E-05	-0.28694E-03	2	2	3
47	-0.30824E-05	3	2	5	0.12890E-05	-0.26677E-03	2	2	3
48	-0.28440E-05	3	2	5	0.11884E-05	-0.24624E-03	2	2	3
49	-0.26400E-05	3	2	5	0.10971E-05	-0.22922E-03	2	2	3
50	-0.24392E-05	3	2	5	0.10129E-05	-0.21159E-03	2	2	3
51	-0.22604E-05	3	2	5	0.93544E-06	-0.19699E-03	2	2	3
52	-0.20989E-05	3	2	5	0.86371E-06	-0.18175E-03	2	2	3
53	-0.19376E-05	3	2	5	0.79856E-06	-0.16863E-03	2	2	3
54	-0.17933E-05	3	2	5	0.73820E-06	-0.15593E-03	2	2	3
55	-0.16619E-05	3	2	5	0.68377E-06	-0.14534E-03	2	2	3
56	-0.15368E-05	3	2	5	0.63123E-06	-0.13388E-03	2	2	3
57	-0.14261E-05	3	2	5	0.58526E-06	-0.12436E-03	2	2	3
58	-0.13230E-05	3	2	5	0.54188E-06	-0.11521E-03	2	2	3
59	-0.12219E-05	3	2	6	0.50186E-06	-0.10648E-03	2	2	3
60	-0.11361E-05	3	2	5	0.46485E-06	-0.98866E-04	2	2	3
61	-0.10483E-05	3	2	6	0.43096E-06	-0.91158E-04	2	2	3
62	-0.97659E-06	3	2	5	0.39992E-06	-0.85472E-04	2	2	3

MAX RESIDAL 1	MAX RESIDAL 2	WORK	REDUCTN/CYCLE	CONV TOLERENCE
0.8248E-03	-0.8547E-04	61.0000	0.9635	0.1000E-05

SECTION CHARACTERISTICS

MACH NO	YAW	ANG OF ATTACK
0.10000	.0	2.0000

SPAN STATION	CL	CD	CM
0.0	.20238	.71088E-02	-.91575E-01
3.87499	.24033	-.24623E-02	-.98798E-01
7.74999	.26515	-.48115E-02	-.10506
11.62498	.28127	-.57074E-02	-.10920
15.49998	.28695	-.65491E-02	-.10935
19.37497	.24168	-.70588E-02	-.86038E-01

WING CHARACTERISTICS

MACH NO	YAW	ANG OF ATTACK			
0.10000	.0	2.0000			
CL	CD FORM	CD FRICTION	CD	L/D FORM	L/D
0.25575	-.32534E-02	.10000E-01	.67466E-02	-78.612	37.
CM YAW	CM ROLL	CM PITCH			
-0.00467	.23919	-.33140			

```
ITERATIVE SOLUTION
    MACH NO       YAW                      ANG OF ATTACK
    0.10000        .0                      2.00000
    NX                NY                         NZ
     80                8                         16
  RELAX FCT 1    RELAX FCT 2           RELAX FCT 3
    1.30000        .70000                1.00000
ITERATION     MAX CORRECN     I    J    K       AVG CORRECN      MAX RESIDAL     I    J    K
        1      0.14421E-02   64    9    4       0.10146E-03     -0.16464E-02     2    8   14
        2     -0.12177E-02    2    9   13       0.74633E-04     -0.12200E-02     2    7    3
        3      0.58530E-03    2    9   13       0.55295E-04      0.10263E-02    80    7    3
        4     -0.31494E-03    2    8   12       0.42176E-04      0.84328E-03    80    7    3
        5     -0.22588E-03    2    8   11       0.36029E-04      0.72680E-03    80    7    3
        6     -0.18484E-03    2    7    8       0.31675E-04      0.63685E-03    80    7    3
        7     -0.16431E-03    2    7    7       0.28676E-04     -0.58917E-03     2    7    3
        8     -0.14915E-03    2    7    6       0.26309E-04     -0.57275E-03     2    7    3
        9     -0.13681E-03    2    7    5       0.24610E-04     -0.53678E-03     2    7    3
       10     -0.12680E-03    2    7    4       0.23100E-04     -0.50334E-03     2    7    3
       11     -0.11748E-03    2    7    3       0.21811E-04     -0.46433E-03     2    7    3
       12     -0.10838E-03    2    7    3       0.20622E-04     -0.42648E-03     2    7    3
       13     -0.98990E-04    2    7    3       0.19538E-04     -0.40392E-03     2    6    3
       14     -0.92368E-04    2    7    3       0.18529E-04     -0.38479E-03     2    6    3
       15     -0.85136E-04    2    7    3       0.17596E-04     -0.36746E-03     2    6    3
       16     -0.78993E-04    2    7    3       0.16727E-04     -0.34705E-03     2    6    3
       17     -0.73568E-04    2    6    4       0.15917E-04     -0.32833E-03     2    6    3
       18     -0.69470E-04    2    6    4       0.15159E-04     -0.30978E-03     2    6    3
       19     -0.65599E-04    2    6    4       0.14450E-04     -0.29637E-03     2    5    3
       20     -0.61961E-04    2    6    4       0.13783E-04     -0.28720E-03     2    5    3
 MAX RESIDAL 1   MAX RESIDAL 2              WORK       REDUCTN/CYCLE  CONV TOLERENCE
  -0.1646E-02     -0.2872E-03               19.0000        0.9122       0.1000E-05
```

SECTION CHARACTERISTICS

MACH NO	YAW	ANG OF ATTACK	
0.10000	.0	2.0000	
SPAN STATION	CL	CC	CM
0.0	.22093	.18197E-01	-.10566
1.93750	.24319	.53303E-02	-.10667
3.87499	.26093	.14194E-02	-.10952
5.81249	.27508	.44517E-05	-.11248
7.74999	.28683	-.72064E-03	-.11524
9.68749	.29643	-.11994E-02	-.11750
11.62499	.30399	-.16262E-02	-.11920
13.56248	.30863	-.20637E-02	-.11986
15.49998	.30746	-.27385E-02	-.11812
17.43747	.29141	-.41059E-02	-.10990
19.37497	.21866	-.42003E-02	-.78297E-01

WING CHARACTERISTICS

MACH NO	YAW	ANG OF ATTACK			
0.10000	.0	2.0000			
CL	CD FORM	CD FRICTION	CD	L/D FORM	L/D
0.27663	.87151E-03	.10000E-01	.10872E-01	317.42	25.
CM YAW	CM ROLL	CM PITCH			
-0.00145	.25615	-.35963			

```
ITERATIVE SCLUTICN
    MACH NO      YAW                    ANG OF ATTACK
    0.1C000      .0                     2.0000
    NX                   NY                     NZ
    160                  16                     32
  RELAX FCT 1    RELAX FCT 2         RELAX FCT 3
    1.20000       .70000                1.0000
ITERATICN     MAX CORRECN    I    J    K     AVG CCRRECN     MAX RESIDAL     I    J    K
        1     0.95314E-03   29   17   24     C.20565E-04    -0.13855E-02     2   16   24
        2    -0.96428E-03    2   17   23     C.12764E-04     0.98673E-03   160   15    3
        3     C.42692E-03    2   17   23     0.96247E-05     0.87373E-03   160   15    3
        4    -0.24824E-03    2   15   15     C.79284E-05     0.73407E-03   160   15    3
        5    -0.21125E-03    2   15   16     0.71493E-05     0.62240E-03   160   15    3
 MAX RESIDAL 1   MAX RESIDAL 2           WORK      REDUCTN/CYCLE  CONV TOLERENCE
   -0.1385E-C2      0.6224E-03           4.000C          0.8187     0.1000E-05
```

```
SECTION CHARACTERISTICS
    MACH NO      YAW                      ANG OF ATTACK
    0.10000      .0                       2.0000
SPAN STATION          CL                      CD              CM
    0.0            .22536              .25222E-01       -.11051
    0.96875        .23679              .13934E-01       -.10917
    1.93750        .24748              .81939E-02       -.10974
    2.90625        .25693              .57553E-02       -.11099
    3.87499        .26526              .42539E-02       -.11244
    4.84374        .27275              .34226E-02       -.11393
    5.81249        .27957              .28496E-02       -.11546
    6.78124        .28574              .24530E-02       -.11686
    7.74999        .29139              .21550E-02       -.11821
    8.71874        .29653              .19061E-02       -.11944
    9.68749        .30116              .16846E-02       -.12056
   10.65624        .30522              .14920E-02       -.12147
   11.62499        .30877              .12848E-02       -.12229
   12.59373        .31151              .10890E-02       -.12276
   13.56248        .31326              .83827E-03       -.12287
   14.53123        .31355              .58364E-03       -.12235
   15.49998        .31168              .12256E-03       -.12093
   16.46872        .30595             -.30467E-03       -.11776
   17.43747        .29369             -.14695E-02       -.11177
   18.40622        .26619             -.49591E-03       -.99299E-01
   19.37497        .20095             -.68273E-04       -.76890E-01

WING CHARACTERISTICS
    MACH NO      YAW                      ANG OF ATTACK
    0.10000      .0                       2.0000
      CL         CD FORM                  CD FRICTION         CD              L/D FORM       L/D
    0.28117      .38318E-02               .10000E-01      .13832E-01        73.379         20.
     CM YAW      CM ROLL                  CM PITCH
    0.00116      .25975                  -.36670

A46C MODIFIED FROM FLO27 OF ANTONY JAMESON,COURANT INSTITUTE
THREE DIMENSIONAL WING ANALYSIS IN TRANSONIC FLOW USING FINITE VOLUME SCHEME
BOEING   VERSION, PREPARED BY DR. HAI-CHOW CHEN STANDARD BOEING   INPUT FORMAT
 END OF CALCULATION
```

SPAN STATION = 0.0
NO. OF DATA POINTS = 101

X/C	CF	X/C	CP	X/C	CP	X/C	CF
1.000000	0.237192	0.960492	0.196593	0.921777	0.143051	0.883857	0.10585
0.846730	0.078746	0.810396	0.055858	0.774854	0.034877	0.740105	0.01716
0.706152	0.007221	0.672998	0.005722	0.640641	0.008583	0.609077	0.00762
0.578298	-0.001550	0.548303	-0.017881	0.519099	-0.033259	0.490693	-0.04034
0.463090	-0.038258	0.436282	-0.032961	0.410261	-0.032450	0.385015	-0.04180
0.360542	-0.060456	0.336847	-0.082953	0.313940	-0.102878	0.291831	-0.11491
0.270527	-0.117293	0.250026	-0.112355	0.230321	-0.105321	0.211398	-0.10458
0.193235	-0.117293	0.175832	-0.140735	0.159224	-0.159315	0.143458	-0.15672
0.128543	-0.133080	0.114441	-0.105236	0.101106	-0.092183	0.088501	-0.10183
0.076658	-0.124054	0.065638	-0.137789	0.055508	-0.120929	0.046334	-0.06766
0.038050	0.001907	0.030635	0.091553	0.024313	0.205585	0.018626	0.28106
0.013378	0.307764	0.009185	0.357900	0.006189	0.464439	0.003596	0.55299
0.001527	0.614166	0.000362	0.678199	0.0	0.710760	0.000363	0.67343
0.001528	0.623294	0.003595	0.589234	0.006187	0.533785	0.009017	0.45067
0.012560	0.363622	0.017041	0.282015	0.022399	0.206130	0.028618	0.13828
0.035734	0.066757	0.043503	-0.044073	0.052008	-0.187695	0.061694	-0.28768
0.072512	-0.310923	0.084319	-0.286460	0.096972	-0.255602	0.110348	-0.24804
0.124444	-0.267506	0.139316	-0.300152	0.155014	-0.329758	0.171564	-0.34829
0.188945	-0.359152	0.207137	-0.369591	0.226132	-0.382296	0.245939	-0.39440
0.266567	-0.402161	0.288015	-0.404750	0.310274	-0.403506	0.333338	-0.39572
0.357204	-0.394166	0.381869	-0.387558	0.407326	-0.380661	0.433576	-0.37387
0.460617	-0.367241	0.488449	-0.360583	0.517071	-0.353498	0.546485	-0.34451
0.576692	-0.330738	0.607694	-0.309289	0.639486	-0.280627	0.672056	-0.25015
0.705395	-0.224343	0.739502	-0.203890	0.774388	-0.180457	0.810059	-0.14323
0.846511	-0.091170	0.883730	-0.035073	0.921712	0.026294	0.960470	0.11575
1.000000	0.211307						

SPAN STATION = 0.96875
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CF
1.000000	0.235694	0.960492	0.204223	0.921777	0.155994	0.883857	0.12070
0.846730	0.094005	0.810396	0.070572	0.774854	0.048637	0.740105	0.02956
0.706152	0.017166	0.672998	0.014850	0.640641	0.017166	0.609077	0.01675
0.578298	0.008174	0.548303	-0.008166	0.519099	-0.024404	0.490693	-0.03280
0.463090	-0.031676	0.436282	-0.026464	0.410261	-0.025272	0.385015	-0.03397
0.360542	-0.052742	0.336847	-0.076669	0.313940	-0.099395	0.291831	-0.11515
0.270527	-0.121466	0.250026	-0.120129	0.230322	-0.116135	0.211398	-0.11834
0.193235	-0.134928	0.175832	-0.164083	0.159224	-0.190139	0.143458	-0.19519
0.128542	-0.177690	0.114441	-0.154436	0.101106	-0.145921	0.088501	-0.16173
0.076658	-0.192344	0.065638	-0.215292	0.055507	-0.205593	0.046334	-0.15487
0.038050	-0.083455	0.030635	0.012943	0.024313	0.138283	0.018626	0.22370
0.013378	0.256675	0.009185	0.315394	0.006189	0.432423	0.003596	0.53092
0.001527	0.599861	0.000362	0.669616	0.0	0.700678	0.000363	0.65476
0.001528	0.596046	0.003595	0.555856	0.006187	0.491687	0.009017	0.39754
0.012560	0.296865	0.017041	0.200817	0.022399	0.111580	0.028618	0.03147

0.035734	-0.051941	0.043503	-0.177332	0.052008	-0.334910	0.061694	-0.43746
0.072512	-0.450552	0.084319	-0.411025	0.096972	-0.367003	0.110347	-0.35076
0.124444	-0.364866	0.139315	-0.392795	0.155014	-0.416500	0.171564	-0.42777
0.188945	-0.430925	0.207137	-0.433930	0.226132	-0.439516	0.245939	-0.44445
0.266567	-0.444812	0.288014	-0.440053	0.310273	-0.431819	0.333338	-0.42164
0.357204	-0.410344	0.381868	-0.398653	0.407327	-0.387260	0.433576	-0.37646
0.460617	-0.366049	0.488449	-0.355763	0.517071	-0.345017	0.546485	-0.33216
0.576692	-0.314431	0.607694	-0.289261	0.639486	-0.257679	0.672056	-0.22523
0.705395	-0.198058	0.739502	-0.175995	0.774388	-0.150383	0.810059	-0.11110
0.846511	-0.058370	0.883730	-0.003517	0.921712	0.053951	0.960470	0.13882
1.000000	0.214168						

SPAN STATION = 1.93750
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.235694	0.960492	0.206130	0.921777	0.159264	0.883857	0.12588
0.846730	0.100136	0.810396	0.077248	0.774854	0.055858	0.740105	0.03678
0.706152	0.024387	0.672998	0.021526	0.640641	0.023433	0.609077	0.02288
0.578298	0.013896	0.548303	-0.002469	0.519099	-0.018980	0.490694	-0.02780
0.463090	-0.027180	0.436282	-0.022496	0.410261	-0.021841	0.385015	-0.03102
0.360542	-0.050332	0.336848	-0.074966	0.313940	-0.098595	0.291831	-0.11548
0.270527	-0.123109	0.250026	-0.123109	0.230322	-0.120546	0.211398	-0.12423
0.193235	-0.142549	0.175832	-0.173705	0.159224	-0.201779	0.143458	-0.20868
0.128543	-0.192727	0.114441	-0.170904	0.101106	-0.164057	0.088501	-0.18192
0.076658	-0.214849	0.065638	-0.239943	0.055508	-0.231641	0.046334	-0.18132
0.038050	-0.109613	0.030635	-0.011946	0.024313	0.115395	0.018626	0.20272
0.013378	0.236647	0.009185	0.297274	0.006189	0.417164	0.003596	0.51852
0.001527	0.589779	0.000362	0.658580	0.0	0.680651	0.000363	0.62274
0.001528	0.556265	0.003595	0.514167	0.006187	0.448772	0.009017	0.34931
0.012560	0.242914	0.017041	0.142642	0.022399	0.050545	0.028618	-0.03137
0.035734	-0.116314	0.043503	-0.245597	0.052008	-0.409033	0.061694	-0.51366
0.072512	-0.523090	0.084319	-0.476948	0.096972	-0.426429	0.110348	-0.40552
0.124444	-0.416798	0.139316	-0.442701	0.155014	-0.464141	0.171564	-0.47243
0.188946	-0.472197	0.207137	-0.471779	0.226132	-0.474036	0.245939	-0.47567
0.266567	-0.472580	0.288015	-0.464286	0.310273	-0.452544	0.333338	-0.43892
0.357204	-0.424291	0.381869	-0.409424	0.407326	-0.395000	0.433577	-0.38128
0.460617	-0.368135	0.488450	-0.355193	0.517071	-0.341901	0.546485	-0.32663
0.576692	-0.306606	0.607694	-0.279410	0.639486	-0.246108	0.672056	-0.21228
0.705395	-0.183974	0.739503	-0.161077	0.774388	-0.135022	0.810059	-0.09599
0.846511	-0.044431	0.883730	0.008174	0.921712	0.062943	0.960471	0.14359
1.000000	0.215122						

SPAN STATION = 2.90625
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.236239	0.960492	0.207084	0.921777	0.161171	0.883857	0.12833
0.846730	0.102997	0.810396	0.080654	0.774854	0.059673	0.740105	0.04100
0.706152	0.029155	0.672998	0.025749	0.640641	0.027657	0.609077	0.02724

0.578297	0.018120	0.548303	0.001499	0.519099	-0.015020	0.490693	-0.02404
0.463090	-0.023663	0.436282	-0.019252	0.410261	-0.018661	0.385015	-0.02831
0.360542	-0.047854	0.336848	-0.072794	0.313940	-0.096747	0.291831	-0.11396
0.270527	-0.121943	0.250026	-0.122326	0.230321	-0.120188	0.211398	-0.12426
0.193235	-0.142966	0.175832	-0.174471	0.159224	-0.202877	0.143458	-0.20999
0.128543	-0.194158	0.114441	-0.172394	0.101106	-0.165514	0.088501	-0.18325
0.076658	-0.215888	0.065638	-0.240505	0.055508	-0.231521	0.046334	-0.18045
0.038050	-0.107944	0.030635	-0.009596	0.024313	0.118256	0.018626	0.20653
0.013378	0.242369	0.009185	0.303950	0.006189	0.423431	0.003596	0.52424
0.001527	0.594548	0.000362	0.657218	0.0	0.668662	0.000363	0.60081
0.001528	0.531878	0.003595	0.490734	0.006187	0.425339	0.009017	0.32356
0.012560	0.214168	0.017041	0.111580	0.022399	0.018120	0.028618	-0.06442
0.035734	-0.150297	0.043503	-0.282083	0.052008	-0.449155	0.061694	-0.55537
0.072512	-0.563034	0.084319	-0.513460	0.096972	-0.459501	0.110348	-0.43628
0.124444	-0.446234	0.139315	-0.471243	0.155014	-0.491636	0.171564	-0.49850
0.188945	-0.496541	0.207137	-0.494378	0.226131	-0.494940	0.245939	-0.49478
0.266567	-0.489856	0.288015	-0.479596	0.310273	-0.465836	0.333338	-0.45019
0.357204	-0.433598	0.381868	-0.416798	0.407326	-0.400501	0.433576	-0.38502
0.460617	-0.370162	0.488449	-0.355610	0.517071	-0.340794	0.546485	-0.32413
0.576692	-0.302860	0.607694	-0.274530	0.639486	-0.240300	0.672056	-0.20576
0.705395	-0.177009	0.739502	-0.153899	0.774388	-0.127929	0.810059	-0.08935
0.846511	-0.038590	0.883730	0.012943	0.921712	0.066348	0.960470	0.14550
1.000000	0.216075						

SPAN STATION = 3.87499
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CF
1.000000	0.236647	0.960491	0.208037	0.921777	0.162125	0.883857	0.12970
0.846730	0.104904	0.810396	0.082970	0.774854	0.061989	0.740105	0.04346
0.706152	0.031471	0.672998	0.028610	0.640641	0.031063	0.609077	0.03010
0.578297	0.020981	0.548303	0.004360	0.519099	-0.011981	0.490693	-0.02104
0.463090	-0.020742	0.436282	-0.016451	0.410260	-0.016153	0.385015	-0.02569
0.360542	-0.045325	0.336848	-0.070291	0.313940	-0.094303	0.291831	-0.11163
0.270527	-0.119737	0.250026	-0.120248	0.230321	-0.118222	0.211398	-0.12241
0.193235	-0.141186	0.175832	-0.172692	0.159224	-0.201063	0.143458	-0.20818
0.128543	-0.192344	0.114440	-0.170512	0.101106	-0.163487	0.088501	-0.18090
0.076658	-0.213061	0.065637	-0.236962	0.055507	-0.227289	0.046334	-0.17557
0.038050	-0.102401	0.030635	-0.003576	0.024313	0.124523	0.018626	0.21321
0.013378	0.250953	0.009184	0.313486	0.006189	0.432423	0.003596	0.53337
0.001527	0.602722	0.000362	0.660488	0.0	0.661986	0.000362	0.58405
0.001528	0.511306	0.003595	0.471251	0.006187	0.406265	0.009017	0.30299
0.012560	0.191280	0.017041	0.087329	0.022399	-0.006974	0.028618	-0.09040
0.035734	-0.176557	0.043503	-0.309791	0.052008	-0.479238	0.061694	-0.58624
0.072512	-0.592249	0.084319	-0.539797	0.096972	-0.483079	0.110347	-0.45789
0.124444	-0.466721	0.139315	-0.490955	0.155014	-0.510514	0.171564	-0.51628
0.188946	-0.513077	0.207137	-0.509688	0.226132	-0.509092	0.245939	-0.50778
0.266567	-0.501624	0.288015	-0.490095	0.310273	-0.475049	0.333338	-0.45810
0.357204	-0.440231	0.381868	-0.422214	0.407326	-0.404750	0.433576	-0.38815

0.460617	-0.372240	0.488449	-0.356717	0.517071	-0.341032	0.546485	-0.32354
0.576692	-0.301548	0.607694	-0.272598	0.639486	-0.237831	0.672056	-0.20293
0.705395	-0.173943	0.739502	-0.150774	0.774388	-0.124863	0.810059	-0.08646
0.846511	-0.036027	0.883730	0.014850	0.921712	0.068256	0.960470	0.14686
1.000000	0.217029						

SPAN STATION = 4.84374
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237192	0.960492	0.208446	0.921777	0.163078	0.883857	0.13065
0.846730	0.105858	0.810396	0.083923	0.774854	0.063487	0.740105	0.04536
0.706152	0.033379	0.672998	0.030518	0.640641	0.032970	0.609077	0.03242
0.578298	0.022888	0.548303	0.006676	0.519099	-0.009562	0.490694	-0.01653
0.463090	-0.018239	0.436282	-0.013947	0.410261	-0.013675	0.385015	-0.02318
0.360542	-0.042728	0.336848	-0.067609	0.313941	-0.091536	0.291831	-0.10881
0.270527	-0.116851	0.250026	-0.117327	0.230321	-0.115241	0.211398	-0.11935
0.193235	-0.137934	0.175832	-0.169175	0.159224	-0.197249	0.143458	-0.20410
0.128543	-0.188053	0.114441	-0.166050	0.101106	-0.158633	0.088501	-0.17545
0.076658	-0.206717	0.065637	-0.229674	0.055507	-0.219107	0.046334	-0.16673
0.038050	-0.093051	0.030635	0.005722	0.024313	0.133514	0.018626	0.22275
0.013378	0.261988	0.009184	0.324930	0.006189	0.443050	0.003596	0.54291
0.001527	0.610352	0.000362	0.662940	0.0	0.655311	0.000362	0.56961
0.001528	0.495093	0.003595	0.456401	0.006186	0.391824	0.009017	0.28773
0.012560	0.174114	0.017041	0.069209	0.022399	-0.025596	0.028618	-0.10919
0.035734	-0.195529	0.043503	-0.329758	0.052008	-0.500977	0.061694	-0.60850
0.072512	-0.613255	0.084319	-0.558666	0.096972	-0.499938	0.110347	-0.47335
0.124444	-0.481384	0.139316	-0.505073	0.155014	-0.524044	0.171564	-0.52903
0.188946	-0.524989	0.207137	-0.520740	0.226132	-0.519344	0.245939	-0.51726
0.266567	-0.510275	0.288015	-0.497852	0.310273	-0.481921	0.333338	-0.46414
0.357205	-0.445434	0.381869	-0.426582	0.407327	-0.408351	0.433577	-0.39107
0.460618	-0.374505	0.488450	-0.358377	0.517072	-0.342105	0.546485	-0.32413
0.576692	-0.301668	0.607694	-0.272325	0.639486	-0.237235	0.672056	-0.20207
0.705395	-0.172896	0.739503	-0.149667	0.774388	-0.123731	0.810059	-0.08536
0.846511	-0.034988	0.883730	0.016212	0.921712	0.068665	0.960471	0.14782
1.000000	0.217983						

SPAN STATION = 5.81249
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237192	0.960492	0.208991	0.921777	0.163623	0.883857	0.13160
0.846730	0.106812	0.810396	0.085422	0.774854	0.064850	0.740105	0.04673
0.706152	0.034877	0.672998	0.032016	0.640642	0.034332	0.609077	0.03392
0.578298	0.024796	0.548303	0.008583	0.519099	-0.007570	0.490693	-0.01645
0.463090	-0.016093	0.436282	-0.011768	0.410261	-0.011444	0.385015	-0.02088
0.360542	-0.040318	0.336848	-0.065046	0.313940	-0.088819	0.291831	-0.10595
0.270527	-0.113904	0.250026	-0.114288	0.230322	-0.112116	0.211398	-0.11607
0.193235	-0.134451	0.175832	-0.165360	0.159225	-0.193119	0.143458	-0.19966
0.128543	-0.183438	0.114441	-0.161196	0.101106	-0.153456	0.088501	-0.16965

0.076658	-0.200084	0.065638	-0.222112	0.055508	-0.210711	0.046335	-0.15777
0.038050	-0.083634	0.030635	0.015259	0.024313	0.142642	0.018626	0.23228
0.013378	0.272478	0.009185	0.335966	0.006189	0.452995	0.003596	0.55204
0.001528	0.618526	0.000362	0.665801	0.0	0.649044	0.000363	0.55585
0.001528	0.479289	0.003595	0.442505	0.006186	0.379017	0.009017	0.27384
0.012560	0.158855	0.017041	0.053406	0.022399	-0.041570	0.028618	-0.12510
0.035734	-0.211332	0.043503	-0.346269	0.052008	-0.518688	0.061694	-0.62647
0.072512	-0.630038	0.084320	-0.573678	0.096973	-0.513222	0.110347	-0.48545
0.124444	-0.492794	0.139316	-0.516048	0.155014	-0.534500	0.171564	-0.53890
0.188546	-0.534177	0.207137	-0.529298	0.226132	-0.527314	0.245939	-0.52463
0.266567	-0.517028	0.288015	-0.504034	0.310274	-0.487481	0.333338	-0.46507
0.357205	-0.449777	0.381869	-0.430388	0.407327	-0.411655	0.433577	-0.39386
0.460618	-0.376855	0.488450	-0.360310	0.517072	-0.343680	0.546485	-0.32535
0.576692	-0.302596	0.607694	-0.272955	0.639486	-0.237593	0.672056	-0.20223
0.705395	-0.172896	0.739502	-0.149548	0.774388	-0.123552	0.810059	-0.08509
0.846511	-0.034630	0.883730	0.016212	0.921712	0.069209	0.960470	0.14836
1.000000	0.218528						

SPAN STATION = 6.78124
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237601	0.960492	0.208991	0.921777	0.164168	0.883857	0.13215
0.846730	0.107356	0.810396	0.085831	0.774854	0.065395	0.740105	0.04768
0.706152	0.036240	0.672958	0.033379	0.640641	0.035831	0.609077	0.03528
0.578298	0.026294	0.548303	0.010082	0.519059	-0.005841	0.490653	-0.01462
0.463090	-0.014211	0.436282	-0.009835	0.410261	-0.009443	0.385015	-0.01877
0.360542	-0.038053	0.336847	-0.062602	0.313940	-0.086197	0.291831	-0.10317
0.270527	-0.110984	0.250025	-0.111248	0.230321	-0.108957	0.211398	-0.11273
0.193235	-0.130815	0.175832	-0.161401	0.159224	-0.188734	0.143457	-0.19499
0.128542	-0.178524	0.114441	-0.156105	0.101106	-0.147973	0.088501	-0.16358
0.076658	-0.193170	0.065638	-0.214313	0.055507	-0.202162	0.046334	-0.14874
0.038050	-0.074284	0.030635	0.024387	0.024313	0.151634	0.018626	0.24141
0.013378	0.282424	0.009185	0.346048	0.006188	0.462123	0.003596	0.56007
0.001527	0.624657	0.000362	0.667708	0.0	0.643321	0.000362	0.54427
0.001527	0.466347	0.003595	0.430516	0.006186	0.367982	0.009017	0.26239
0.012560	0.146457	0.017041	0.040054	0.022399	-0.054981	0.028618	-0.13841
0.035734	-0.224582	0.043503	-0.360072	0.052008	-0.533521	0.061694	-0.64150
0.072512	-0.644062	0.084319	-0.586127	0.096972	-0.524248	0.110347	-0.49550
0.124444	-0.502280	0.139315	-0.525142	0.155014	-0.543211	0.171564	-0.54710
0.188945	-0.541849	0.207137	-0.536467	0.226131	-0.533998	0.245939	-0.53088
0.266567	-0.522792	0.288015	-0.509296	0.310274	-0.492266	0.333338	-0.47341
0.357204	-0.453668	0.381868	-0.433896	0.407326	-0.414746	0.433576	-0.39663
0.460618	-0.379264	0.488449	-0.362422	0.517072	-0.345528	0.546485	-0.32695
0.576692	-0.303933	0.607694	-0.274020	0.639486	-0.238453	0.672056	-0.20291
0.705395	-0.173432	0.739502	-0.149940	0.774388	-0.123850	0.810059	-0.08527
0.846511	-0.034656	0.883730	0.016757	0.921712	0.069618	0.960470	0.14877
1.000000	0.218936						

SPAN STATION = 7.74999
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237601	0.960492	0.208991	0.921778	0.164168	0.883857	0.13256
0.846730	0.107765	0.810396	0.086784	0.774854	0.066348	0.740105	0.04863
0.706152	0.037193	0.672958	0.034332	0.640642	0.036785	0.609077	0.03624
0.578298	0.027657	0.548304	0.011444	0.519099	-0.004385	0.490693	-0.01307
0.463090	-0.012577	0.436282	-0.008140	0.410261	-0.007655	0.385015	-0.01686
0.360542	-0.036001	0.336848	-0.060397	0.313941	-0.083813	0.291831	-0.10058
0.270527	-0.108242	0.250026	-0.108395	0.230321	-0.105977	0.211398	-0.10957
0.193235	-0.127426	0.175832	-0.157620	0.159224	-0.184630	0.143457	-0.19058
0.128543	-0.173909	0.114441	-0.151251	0.101106	-0.142787	0.088501	-0.15783
0.076658	-0.186656	0.065638	-0.206930	0.055508	-0.194064	0.046334	-0.14023
0.038050	-0.065437	0.030635	0.033379	0.024313	0.159809	0.018626	0.24959
0.013378	0.292097	0.009185	0.355993	0.006188	0.470706	0.003596	0.56770
0.001527	0.630924	0.000362	0.669207	0.0	0.637599	0.000363	0.53242
0.001528	0.453540	0.003595	0.419617	0.006186	0.357492	0.009017	0.25149
0.012560	0.134468	0.017041	0.028202	0.022399	-0.066927	0.028618	-0.15017
0.035734	-0.236188	0.043503	-0.372035	0.052008	-0.546277	0.061694	-0.65430
0.072512	-0.655941	0.084320	-0.596677	0.096972	-0.533555	0.110348	-0.50394
0.124444	-0.510250	0.139316	-0.532746	0.155014	-0.550491	0.171564	-0.55400
0.188946	-0.548295	0.207137	-0.542496	0.226131	-0.539643	0.245939	-0.53616
0.266567	-0.527757	0.288015	-0.513877	0.310274	-0.496481	0.333338	-0.47728
0.357205	-0.457210	0.381869	-0.437081	0.407327	-0.417658	0.433576	-0.39924
0.460618	-0.381640	0.488450	-0.364533	0.517072	-0.347435	0.546485	-0.32862
0.576692	-0.305423	0.607694	-0.275331	0.639486	-0.239560	0.672056	-0.20380
0.705395	-0.174182	0.739503	-0.150561	0.774388	-0.124352	0.810059	-0.08563
0.846511	-0.034894	0.883730	0.016212	0.921712	0.069618	0.960470	0.14877
1.000000	0.219481						

SPAN STATION = 8.71874
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237601	0.960491	0.209400	0.921777	0.164577	0.883857	0.13256
0.846730	0.108310	0.810396	0.087329	0.774854	0.066757	0.740105	0.04918
0.706152	0.037738	0.672998	0.035286	0.640641	0.037738	0.609077	0.03719
0.578298	0.028610	0.548303	0.012943	0.519099	-0.003099	0.490693	-0.01168
0.463090	-0.011112	0.436282	-0.006616	0.410261	-0.006054	0.385015	-0.01514
0.360542	-0.034119	0.336848	-0.058344	0.313941	-0.081582	0.291831	-0.09820
0.270527	-0.105679	0.250026	-0.105679	0.230321	-0.103116	0.211398	-0.10657
0.193235	-0.124148	0.175832	-0.154018	0.159225	-0.180670	0.143458	-0.18632
0.128542	-0.169473	0.114441	-0.146602	0.101106	-0.137814	0.088501	-0.15232
0.076658	-0.180457	0.065638	-0.199965	0.055508	-0.186477	0.046334	-0.13228
0.038050	-0.057246	0.030635	0.041553	0.024313	0.167438	0.018626	0.25722
0.013378	0.300680	0.009185	0.364167	0.006189	0.478336	0.003596	0.57452
0.001527	0.635692	0.000362	0.670161	0.0	0.631877	0.000363	0.52179
0.001528	0.442096	0.003595	0.410080	0.006187	0.348909	0.009017	0.24236
0.012560	0.124523	0.017041	0.017711	0.022400	-0.077503	0.028618	-0.16060

0.035734	-0.246465	0.043503	-0.382653	0.052008	-0.557601	0.061695	-0.66571
0.072512	-0.666550	0.084320	-0.606060	0.096972	-0.541849	0.110348	-0.51149
0.124444	-0.517351	0.139316	-0.539584	0.155014	-0.557031	0.171564	-0.56018
0.188946	-0.554093	0.207137	-0.547911	0.226132	-0.544761	0.245939	-0.54098
0.266567	-0.532244	0.288015	-0.518032	0.310274	-0.500321	0.333338	-0.48082
0.357205	-0.460454	0.381869	-0.440053	0.407327	-0.420366	0.433577	-0.40171
0.460618	-0.383905	0.488449	-0.366620	0.517071	-0.349309	0.546485	-0.33032
0.576692	-0.306938	0.607694	-0.276668	0.639486	-0.240718	0.672056	-0.20478
0.705395	-0.175008	0.739502	-0.151277	0.774388	-0.124948	0.810059	-0.08611
0.846511	-0.035192	0.883730	0.016212	0.921712	0.069618	0.960470	0.14931
1.000000	0.219890						

SPAN STATION = 9.68749
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237601	0.960492	0.209400	0.921777	0.164577	0.883857	0.13310
0.846730	0.108719	0.810396	0.087738	0.774854	0.067302	0.740105	0.04959
0.706152	0.038692	0.672998	0.035831	0.640642	0.038692	0.609077	0.03814
0.578297	0.029564	0.548303	0.013896	0.519099	-0.001992	0.490693	-0.01051
0.463090	-0.009860	0.436282	-0.005305	0.410261	-0.004683	0.385015	-0.01364
0.360542	-0.032485	0.336847	-0.056556	0.313940	-0.079589	0.291831	-0.09605
0.270527	-0.103414	0.250026	-0.103295	0.230321	-0.100613	0.211398	-0.10386
0.193235	-0.121227	0.175832	-0.150800	0.159224	-0.177094	0.143457	-0.18248
0.128542	-0.165420	0.114441	-0.142404	0.101106	-0.133319	0.088501	-0.14731
0.076658	-0.174769	0.065637	-0.193562	0.055507	-0.179512	0.046334	-0.12492
0.038050	-0.049702	0.030635	0.049182	0.024312	0.174659	0.018626	0.26430
0.013378	0.308309	0.009184	0.372342	0.006188	0.485012	0.003596	0.58024
0.001527	0.639915	0.000362	0.670569	0.0	0.626155	0.000362	0.51130
0.001528	0.431061	0.003595	0.400407	0.006186	0.340325	0.009017	0.23337
0.012560	0.114986	0.017041	0.007629	0.022399	-0.087210	0.028618	-0.17006
0.035734	-0.255746	0.043503	-0.392173	0.052008	-0.567675	0.061694	-0.67578
0.072512	-0.675840	0.084319	-0.614294	0.096972	-0.549095	0.110347	-0.51806
0.124444	-0.523533	0.139315	-0.545502	0.155014	-0.562676	0.171564	-0.56552
0.188946	-0.559109	0.207137	-0.552637	0.226131	-0.549214	0.245939	-0.54514
0.266567	-0.536169	0.288015	-0.521719	0.310273	-0.503736	0.333338	-0.48393
0.357204	-0.463367	0.381868	-0.442726	0.407327	-0.422835	0.433576	-0.40397
0.460617	-0.385983	0.488450	-0.368527	0.517071	-0.351063	0.546485	-0.33193
0.576692	-0.308335	0.607694	-0.277919	0.639486	-0.241790	0.672056	-0.20573
0.705395	-0.175817	0.739502	-0.151966	0.774388	-0.125519	0.810059	-0.08652
0.846511	-0.035465	0.883730	0.016212	0.921712	0.069618	0.960470	0.14931
1.000000	0.219890						

SPAN STATION = 10.65624
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237601	0.960492	0.209400	0.921777	0.164577	0.883857	0.13310
0.846730	0.108719	0.810396	0.087738	0.774854	0.067711	0.740105	0.05013
0.706152	0.039101	0.672998	0.036240	0.640642	0.039101	0.609077	0.03869

0.578298	0.030109	0.548303	0.014305	0.519099	-0.001073	0.490694	-0.00950
0.463090	-0.008787	0.436282	-0.004172	0.410261	-0.003457	0.385015	-0.01233
0.360542	-0.031054	0.336848	-0.054947	0.313941	-0.077827	0.291831	-0.09412
0.270527	-0.101328	0.250026	-0.101089	0.230321	-0.098297	0.211398	-0.10138
0.193235	-0.118520	0.175832	-0.147768	0.159224	-0.173765	0.143458	-0.17891
0.128543	-0.161673	0.114441	-0.138470	0.101106	-0.129121	0.088501	-0.14266
0.076658	-0.169533	0.065638	-0.187669	0.055507	-0.173109	0.046334	-0.11824
0.038050	-0.042822	0.030635	0.055858	0.024313	0.180789	0.018627	0.27098
0.013378	0.315394	0.009184	0.379017	0.006188	0.491278	0.003596	0.58542
0.001527	0.643321	0.000362	0.670569	0.0	0.620842	0.000363	0.50231
0.001528	0.421524	0.003595	0.392369	0.006186	0.332696	0.009017	0.22561
0.012560	0.106812	0.017041	-0.000630	0.022400	-0.095793	0.028618	-0.17852
0.035734	-0.264023	0.043503	-0.400679	0.052008	-0.576700	0.061694	-0.68487
0.072512	-0.684253	0.084320	-0.621719	0.096972	-0.555669	0.110348	-0.52407
0.124444	-0.529187	0.139315	-0.550909	0.155014	-0.567853	0.171564	-0.57040
0.188946	-0.563690	0.207137	-0.556946	0.226132	-0.553259	0.245939	-0.54897
0.266567	-0.539737	0.288014	-0.525023	0.310273	-0.506827	0.333338	-0.48679
0.357205	-0.465981	0.381869	-0.445136	0.407327	-0.425066	0.433577	-0.40602
0.460618	-0.387856	0.488449	-0.370221	0.517072	-0.352604	0.546485	-0.33332
0.576692	-0.309612	0.607694	-0.279026	0.639486	-0.242744	0.672056	-0.20653
0.705395	-0.176498	0.739503	-0.152562	0.774388	-0.125962	0.810059	-0.08688
0.846511	-0.035703	0.883730	0.016212	0.921712	0.069618	0.960471	0.14931
1.000000	0.220435						

SPAN STATION = 11.62499
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237601	0.960492	0.209400	0.921777	0.165122	0.883857	0.13351
0.846730	0.109264	0.810396	0.088283	0.774854	0.068256	0.740105	0.05054
0.706152	0.039101	0.672998	0.036785	0.640642	0.039646	0.609077	0.02910
0.578298	0.031063	0.548303	0.015259	0.519099	-0.000383	0.490694	-0.00872
0.463090	-0.007953	0.436282	-0.003304	0.410261	-0.002529	0.385015	-0.01132
0.360542	-0.029887	0.336848	-0.053636	0.313941	-0.076371	0.291831	-0.09254
0.270527	-0.099633	0.250026	-0.099276	0.230322	-0.096355	0.211398	-0.09933
0.193235	-0.116221	0.175832	-0.145231	0.159224	-0.170929	0.143458	-0.17587
0.128543	-0.158455	0.114441	-0.135081	0.101106	-0.125485	0.088501	-0.13862
0.076658	-0.164943	0.065638	-0.182458	0.055507	-0.167421	0.046334	-0.11226
0.038050	-0.036682	0.030635	0.061580	0.024313	0.186511	0.018627	0.27670
0.013378	0.321661	0.009185	0.385693	0.006188	0.496592	0.003596	0.59018
0.001527	0.646727	0.000362	0.670569	0.0	0.615529	0.000362	0.49318
0.001528	0.411987	0.003595	0.383786	0.006187	0.325475	0.009017	0.21798
0.012560	0.098773	0.017041	-0.008728	0.022399	-0.103806	0.028618	-0.18632
0.035734	-0.271610	0.043503	-0.408411	0.052008	-0.584849	0.061694	-0.69298
0.072512	-0.691703	0.084320	-0.628284	0.096972	-0.561433	0.110348	-0.52927
0.124444	-0.534058	0.139315	-0.555549	0.155014	-0.572273	0.171564	-0.57459
0.188946	-0.567615	0.207137	-0.560599	0.226132	-0.556707	0.245939	-0.55222
0.266567	-0.542768	0.288014	-0.527850	0.310274	-0.509415	0.333338	-0.48920
0.357205	-0.468177	0.381868	-0.447163	0.407327	-0.426880	0.433577	-0.40769

0.460617	-0.389346	0.488450	-0.371584	0.517071	-0.353830	0.546485	-0.33443
0.576692	-0.310566	0.607694	-0.279852	0.639486	-0.243425	0.672056	-0.20707
0.705395	-0.176949	0.739502	-0.152886	0.774388	-0.126234	0.810059	-0.08703
0.846511	-0.035729	0.883730	0.016212	0.921712	0.069618	0.960470	0.14972
1.000000	0.220435						

SPAN STATION = 12.59373
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237601	0.960491	0.209400	0.921777	0.165122	0.883857	0.13351
0.846730	0.109264	0.810396	0.088283	0.774854	0.068256	0.740105	0.05054
0.706151	0.039646	0.672997	0.037193	0.640641	0.039646	0.609077	0.03964
0.578298	0.031063	0.548303	0.015259	0.519099	0.0	0.490693	-0.00822
0.463090	-0.007391	0.436282	-0.002682	0.410261	-0.001848	0.385015	-0.01055
0.360542	-0.028993	0.336848	-0.052622	0.313940	-0.075204	0.291831	-0.09123
0.270527	-0.098203	0.250026	-0.097760	0.230321	-0.094720	0.211398	-0.09752
0.193235	-0.114262	0.175832	-0.143000	0.159224	-0.168434	0.143458	-0.17313
0.128543	-0.155594	0.114441	-0.132067	0.101106	-0.122241	0.088501	-0.13502
0.076658	-0.160839	0.065638	-0.177843	0.055507	-0.162355	0.046334	-0.10699
0.038050	-0.031258	0.030635	0.067302	0.024313	0.191825	0.018626	0.28147
0.013378	0.326838	0.009184	0.390870	0.006189	0.501360	0.003595	0.59359
0.001527	0.649044	0.000362	0.670161	0.0	0.610760	0.000362	0.48555
0.001528	0.403813	0.003595	0.377110	0.006186	0.319208	0.009017	0.21185
0.012560	0.092098	0.017041	-0.015582	0.022399	-0.110592	0.028618	-0.19296
0.035734	-0.278098	0.043503	-0.415070	0.052008	-0.591857	0.061694	-0.70002
0.072512	-0.698209	0.084320	-0.634023	0.096972	-0.566483	0.110347	-0.53385
0.124444	-0.538341	0.139315	-0.559645	0.155014	-0.576198	0.171564	-0.57825
0.188945	-0.571030	0.207137	-0.563775	0.226131	-0.559679	0.245938	-0.55497
0.266567	-0.545323	0.288014	-0.530192	0.310273	-0.511553	0.333338	-0.49113
0.357204	-0.469932	0.381868	-0.448738	0.407326	-0.428276	0.433576	-0.40891
0.460617	-0.390444	0.488449	-0.372512	0.517071	-0.354631	0.546485	-0.33508
0.576692	-0.311102	0.607694	-0.280244	0.639486	-0.243723	0.672056	-0.20725
0.705395	-0.177034	0.739502	-0.152920	0.774388	-0.126175	0.810059	-0.08691
0.846510	-0.035524	0.883730	0.016212	0.921712	0.070163	0.960470	0.15027
1.000000	0.220844						

SPAN STATION = 13.56248
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237601	0.960492	0.209400	0.921777	0.165122	0.883857	0.13351
0.846730	0.109264	0.810396	0.088283	0.774854	0.068256	0.740105	0.05054
0.706152	0.039646	0.672998	0.037193	0.640642	0.039646	0.609077	0.03564
0.578298	0.031063	0.548303	0.015804	0.519099	0.0	0.490694	-0.00804
0.463090	-0.007178	0.436283	-0.002416	0.410261	-0.001550	0.385015	-0.01019
0.360542	-0.028551	0.336847	-0.052060	0.313940	-0.074548	0.291831	-0.09048
0.270527	-0.097368	0.250026	-0.096806	0.230321	-0.093673	0.211398	-0.09638
0.193235	-0.112951	0.175832	-0.141484	0.159224	-0.166706	0.143458	-0.17122
0.128542	-0.153542	0.114441	-0.129896	0.101106	-0.119856	0.088501	-0.13230

0.076658	-0.157680	0.065638	-0.174241	0.055507	-0.158370	0.046334	-0.10275
0.038051	-0.026882	0.030635	0.071526	0.024313	0.195639	0.018626	0.28528
0.013378	0.331742	0.009184	0.395230	0.006188	0.505175	0.003596	0.59700
0.001527	0.650951	0.000362	0.669616	0.0	0.605992	0.000363	0.47792
0.001528	0.396184	0.003595	0.370434	0.006187	0.313078	0.009017	0.20558
0.012560	0.085831	0.017041	-0.022054	0.022399	-0.116936	0.028618	-0.19907
0.035734	-0.283991	0.043503	-0.420988	0.052008	-0.598039	0.061695	-0.70611
0.072512	-0.703761	0.084320	-0.638894	0.096972	-0.570706	0.110347	-0.53762
0.124444	-0.541815	0.139316	-0.562949	0.155014	-0.579263	0.171564	-0.58110
0.188946	-0.573644	0.207137	-0.566185	0.226132	-0.561850	0.245939	-0.55697
0.266567	-0.547103	0.288015	-0.531767	0.310274	-0.512958	0.333338	-0.49232
0.357205	-0.470945	0.381868	-0.449572	0.407326	-0.428932	0.433577	-0.40942
0.460617	-0.390802	0.488450	-0.372750	0.517071	-0.354690	0.546485	-0.33502
0.576693	-0.310923	0.607694	-0.279972	0.639486	-0.243340	0.672056	-0.20677
0.705395	-0.176498	0.739503	-0.152324	0.774388	-0.125544	0.810059	-0.08623
0.846511	-0.034835	0.883730	0.017166	0.921712	0.070572	0.960470	0.15068
1.000000	0.221389						

SPAN STATION = 14.53123
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CF
1.000000	0.238146	0.960492	0.209400	0.921777	0.164577	0.883857	0.13310
0.846730	0.108719	0.810396	0.087738	0.774854	0.067711	0.740105	0.05013
0.706152	0.039101	0.672998	0.036785	0.640642	0.039101	0.609077	0.03910
0.578298	0.031063	0.548304	0.015259	0.519099	-0.000034	0.490694	-0.00822
0.463090	-0.007331	0.436282	-0.002563	0.410261	-0.001635	0.385015	-0.01021
0.360543	-0.028517	0.336848	-0.051941	0.313940	-0.074344	0.291831	-0.09016
0.270527	-0.096951	0.250026	-0.096296	0.230321	-0.093077	0.211398	-0.09569
0.193235	-0.112116	0.175832	-0.140471	0.159225	-0.165479	0.143458	-0.16983
0.128543	-0.152026	0.114441	-0.128261	0.101106	-0.118009	0.088502	-0.13019
0.076658	-0.155210	0.065638	-0.171380	0.055508	-0.155210	0.046334	-0.09942
0.038050	-0.023425	0.030635	0.074932	0.024313	0.198909	0.018626	0.28869
0.013378	0.335012	0.009184	0.398500	0.006189	0.508036	0.003596	0.59931
0.001527	0.652449	0.000362	0.668662	0.0	0.602722	0.000363	0.47220
0.001528	0.390462	0.003595	0.366075	0.006187	0.309263	0.009017	0.20136
0.012560	0.081607	0.017041	-0.026490	0.022399	-0.121321	0.028618	-0.20332
0.035734	-0.288129	0.043503	-0.425185	0.052008	-0.602441	0.061694	-0.71044
0.072512	-0.707771	0.084319	-0.642368	0.096972	-0.573703	0.110348	-0.54029
0.124444	-0.544284	0.139316	-0.565205	0.155015	-0.581367	0.171564	-0.58300
0.188946	-0.575313	0.207137	-0.567615	0.226132	-0.563094	0.245939	-0.55798
0.266567	-0.547937	0.288015	-0.532397	0.310274	-0.513341	0.333338	-0.49250
0.357205	-0.470919	0.381869	-0.449334	0.407327	-0.428549	0.433577	-0.40882
0.460617	-0.390027	0.488450	-0.371822	0.517071	-0.353651	0.546485	-0.33383
0.576693	-0.309646	0.607694	-0.278575	0.639486	-0.241850	0.672056	-0.20526
0.705395	-0.174957	0.739503	-0.150774	0.774388	-0.123995	0.810059	-0.08476
0.846511	-0.033438	0.883730	0.018120	0.921712	0.072070	0.960471	0.15163
1.000000	0.222342						

SPAN STATION = 15.49998
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CF
1.000000	0.238146	0.960492	0.209400	0.921777	0.164577	0.883857	0.13256
0.846730	0.108310	0.810396	0.087329	0.774854	0.067302	0.740105	0.04959
0.706152	0.038692	0.672998	0.035831	0.640641	0.038692	0.609077	0.03869
0.578297	0.030109	0.548303	0.014850	0.519099	-0.000715	0.490693	-0.00890
0.463090	-0.008013	0.436282	-0.003244	0.410261	-0.002350	0.385015	-0.01093
0.360542	-0.029206	0.336847	-0.052597	0.313940	-0.074940	0.291831	-0.09072
0.270527	-0.097488	0.250026	-0.096832	0.230322	-0.093613	0.211398	-0.09617
0.193235	-0.112534	0.175831	-0.140769	0.159224	-0.165692	0.143458	-0.16995
0.128543	-0.152085	0.114440	-0.128227	0.101105	-0.117923	0.088501	-0.12992
0.076658	-0.154674	0.065637	-0.170546	0.055507	-0.154104	0.046334	-0.09817
0.038050	-0.022079	0.030634	0.076294	0.024313	0.199863	0.018626	0.29018
0.013378	0.336511	0.009184	0.400407	0.006188	0.509398	0.003596	0.59986
0.001527	0.652449	0.000362	0.667300	0.0	0.598363	0.000362	0.46634
0.001528	0.384194	0.003595	0.360353	0.006187	0.304495	0.009017	0.19700
0.012560	0.076839	0.017041	-0.030960	0.022399	-0.125578	0.028618	-0.20728
0.035734	-0.291816	0.043503	-0.428694	0.052008	-0.605822	0.061694	-0.71364
0.072512	-0.710564	0.084319	-0.644718	0.096972	-0.575636	0.110348	-0.54187
0.124444	-0.545561	0.139315	-0.566219	0.155014	-0.582082	0.171564	-0.58341
0.188945	-0.575457	0.207137	-0.567462	0.226131	-0.562651	0.245939	-0.55726
0.266567	-0.546932	0.288015	-0.531111	0.310273	-0.511791	0.333338	-0.49071
0.357204	-0.468867	0.381869	-0.447043	0.407327	-0.426045	0.433576	-0.40612
0.460617	-0.387141	0.488449	-0.368765	0.517071	-0.350433	0.546485	-0.33053
0.576692	-0.306223	0.607694	-0.275152	0.639485	-0.238427	0.672056	-0.20189
0.705395	-0.171619	0.739502	-0.147556	0.774388	-0.120929	0.810059	-0.08188
0.846511	-0.030816	0.883730	0.020572	0.921712	0.073978	0.960470	0.15354
1.000000	0.223296						

SPAN STATION = 16.46872
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.237192	0.960491	0.208037	0.921777	0.163078	0.883857	0.13119
0.846730	0.106812	0.810396	0.085831	0.774854	0.065804	0.740105	0.04822
0.706152	0.037193	0.672998	0.034877	0.640642	0.037193	0.609077	0.03719
0.578298	0.029155	0.548303	0.013351	0.519099	-0.001754	0.490694	-0.00992
0.463090	-0.009026	0.436282	-0.004257	0.410261	-0.003363	0.385015	-0.01192
0.360542	-0.030160	0.336847	-0.053491	0.313940	-0.075800	0.291831	-0.09156
0.270527	-0.098297	0.250026	-0.097641	0.230321	-0.094422	0.211398	-0.09695
0.193235	-0.113249	0.175832	-0.141391	0.159224	-0.166195	0.143457	-0.17036
0.128543	-0.152443	0.114441	-0.128559	0.101106	-0.118128	0.088501	-0.12995
0.076658	-0.154461	0.065638	-0.170069	0.055508	-0.153422	0.046334	-0.09730
0.038050	-0.021126	0.030635	0.077248	0.024313	0.200817	0.018626	0.29059
0.013378	0.336919	0.009185	0.400952	0.006189	0.509398	0.003596	0.59931
0.001527	0.651496	0.000362	0.665392	0.0	0.596046	0.000363	0.46443
0.001528	0.383241	0.003595	0.359808	0.006186	0.303950	0.009017	0.19700
0.012560	0.077248	0.017041	-0.030066	0.022399	-0.124506	0.028618	-0.20615

0.035734	-0.290539	0.043503	-0.427118	0.052008	-0.604016	0.061694	-0.71178
0.072512	-0.708725	0.084320	-0.642989	0.096972	-0.573942	0.110347	-0.54009
0.124444	-0.543543	0.139316	-0.563894	0.155014	-0.579468	0.171564	-0.58050
0.188945	-0.572247	0.207137	-0.563928	0.226132	-0.558725	0.245939	-0.55296
0.266567	-0.542257	0.288015	-0.526096	0.310273	-0.506410	0.333338	-0.48501
0.357205	-0.462890	0.381868	-0.440793	0.407326	-0.419532	0.433576	-0.39939
0.460617	-0.380210	0.488449	-0.361706	0.517071	-0.343263	0.546485	-0.32333
0.576692	-0.299079	0.607694	-0.268102	0.639485	-0.231581	0.672056	-0.19529
0.705395	-0.165335	0.739502	-0.141604	0.774388	-0.115386	0.810059	-0.07681
0.846511	-0.026286	0.883730	0.024796	0.921712	0.077248	0.960470	0.15599
1.000000	0.225203						

SPAN STATION = 17.43747
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.235285	0.960492	0.206539	0.921777	0.160762	0.883857	0.12833
0.846730	0.103542	0.810396	0.082561	0.774854	0.062534	0.740105	0.04536
0.706151	0.034332	0.672958	0.022016	0.640642	0.034877	0.609077	0.03487
0.578298	0.026703	0.548303	0.011444	0.519099	-0.003542	0.490654	-0.01156
0.463090	-0.010550	0.436283	-0.005722	0.410261	-0.004794	0.385015	-0.01329
0.360542	-0.031471	0.336848	-0.054768	0.313940	-0.077052	0.291831	-0.09283
0.270527	-0.099693	0.250025	-0.099216	0.230322	-0.096176	0.211398	-0.09895
0.193235	-0.115420	0.175832	-0.143681	0.159225	-0.168613	0.143458	-0.17301
0.128542	-0.155449	0.114441	-0.131922	0.101106	-0.121857	0.088501	-0.13385
0.076658	-0.158429	0.065638	-0.174029	0.055508	-0.157382	0.046334	-0.10150
0.038050	-0.025690	0.030635	0.072070	0.024313	0.194686	0.018626	0.28392
0.013378	0.330244	0.009185	0.394276	0.006189	0.501769	0.003596	0.59114
0.001528	0.643321	0.000362	0.657218	0.0	0.587872	0.000363	0.45490
0.001528	0.372750	0.003595	0.348909	0.006186	0.294958	0.009017	0.18991
0.012560	0.072070	0.017041	-0.034298	0.022400	-0.127571	0.028618	-0.20818
0.035734	-0.291313	0.043503	-0.425867	0.052008	-0.600534	0.061694	-0.70709
0.072512	-0.703880	0.084319	-0.638272	0.096972	-0.569097	0.110347	-0.53471
0.124444	-0.537242	0.139315	-0.556529	0.155014	-0.570995	0.171564	-0.57114
0.188946	-0.562055	0.207137	-0.552961	0.226131	-0.546958	0.245939	-0.54041
0.266567	-0.529034	0.288014	-0.512302	0.310273	-0.492173	0.333338	-0.47040
0.357205	-0.447963	0.381868	-0.425662	0.407326	-0.404247	0.433577	-0.38405
0.460617	-0.364891	0.488450	-0.346448	0.517071	-0.328209	0.546485	-0.30854
0.576692	-0.284706	0.607694	-0.254290	0.639486	-0.218451	0.672056	-0.18296
0.705395	-0.153874	0.739503	-0.131088	0.774388	-0.105917	0.810059	-0.06847
0.846511	-0.019218	0.883730	0.030109	0.921712	0.081062	0.960470	0.15831
1.000000	0.224658						

SPAN STATION = 18.40622
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.224658	0.960494	0.195639	0.921779	0.149727	0.883859	0.11730
0.846732	0.093051	0.810398	0.072479	0.774855	0.052997	0.740107	0.03583
0.706153	0.025340	0.672999	0.023433	0.640643	0.026703	0.609078	0.02724

0.578299	0.020027	0.548304	0.005313	0.519100	-0.008821	0.490695	-0.01611
0.463050	-0.014509	0.436283	-0.009060	0.410261	-0.007357	0.385016	-0.01502
0.360543	-0.032272	0.336848	-0.054585	0.313941	-0.075979	0.291832	-0.09102
0.270527	-0.097283	0.250026	-0.096296	0.230322	-0.092813	0.211398	-0.09504
0.193235	-0.110745	0.175832	-0.138027	0.159224	-0.161912	0.143458	-0.16547
0.128543	-0.147521	0.114440	-0.123875	0.101106	-0.113487	0.088501	-0.12453
0.076658	-0.147351	0.065638	-0.160660	0.055507	-0.141748	0.046334	-0.08456
0.038050	-0.008438	0.030635	0.089237	0.024313	0.208446	0.018626	0.29250
0.013377	0.339372	0.009184	0.404767	0.006188	0.505175	0.003596	0.58542
0.001527	0.631332	0.000362	0.638962	0.0	0.567709	0.000362	0.44250
0.001528	0.366075	0.003595	0.346456	0.006187	0.302042	0.009017	0.20844
0.012560	0.097275	0.017040	-0.005424	0.022399	-0.096628	0.028618	-0.17635
0.035734	-0.257058	0.043503	-0.384910	0.052008	-0.554153	0.061694	-0.66156
0.072512	-0.662540	0.084319	-0.601368	0.096972	-0.535190	0.110347	-0.50177
0.124444	-0.503915	0.139316	-0.522290	0.155014	-0.536135	0.171564	-0.53596
0.188946	-0.526718	0.207138	-0.517385	0.226132	-0.511050	0.245939	-0.50415
0.266567	-0.492530	0.288015	-0.475790	0.310274	-0.455839	0.333339	-0.43437
0.357205	-0.412362	0.381869	-0.390623	0.407327	-0.369923	0.433577	-0.35052
0.460619	-0.332287	0.488450	-0.314908	0.517072	-0.297827	0.546486	-0.27546
0.576693	-0.257032	0.607695	-0.228243	0.639487	-0.194362	0.672057	-0.16116
0.705356	-0.134664	0.739504	-0.114526	0.774389	-0.091919	0.810061	-0.05709
0.846512	-0.010874	0.883731	0.034877	0.921714	0.081062	0.960473	0.15449
1.000000	0.218936						

SPAN STATION = 19.37497
NO. OF DATA POINTS = 101

X/C	CP	X/C	CP	X/C	CP	X/C	CP
1.000000	0.155040	0.960490	0.140190	0.921779	0.105858	0.883856	0.08106
0.846732	0.061989	0.810397	0.045368	0.774855	0.029564	0.740106	0.01580
0.706153	0.006267	0.672999	0.002861	0.640643	0.003406	0.609078	0.00245
0.578299	-0.003576	0.548305	-0.014748	0.519100	-0.026047	0.490694	-0.03230
0.463090	-0.032212	0.436283	-0.029206	0.410261	-0.028576	0.385016	-0.03445
0.360543	-0.047054	0.336849	-0.063351	0.313941	-0.079172	0.291831	-0.09058
0.270527	-0.095793	0.250026	-0.095853	0.230322	-0.094090	0.211399	-0.09644
0.193236	-0.108421	0.175832	-0.128320	0.159224	-0.145205	0.143458	-0.14729
0.128543	-0.134272	0.114441	-0.117804	0.101106	-0.111307	0.088502	-0.11979
0.076658	-0.135047	0.065638	-0.141093	0.055507	-0.121023	0.046334	-0.07237
0.038050	-0.009741	0.030635	0.077248	0.024313	0.177928	0.018626	0.23950
0.013378	0.283377	0.009184	0.363622	0.006188	0.458717	0.003596	0.52179
0.001527	0.553404	0.000362	0.532831	0.0	0.426292	0.000362	0.28487
0.001528	0.206130	0.003595	0.202315	0.006186	0.200817	0.009017	0.13978
0.012560	0.035286	0.017041	-0.060397	0.022399	-0.136954	0.028618	-0.20044
0.035734	-0.255985	0.043503	-0.339849	0.052008	-0.460931	0.061694	-0.54178
0.072512	-0.542019	0.084319	-0.494403	0.096972	-0.440436	0.110347	-0.40853
0.124444	-0.402221	0.139316	-0.408411	0.155014	-0.412396	0.171564	-0.40733
0.188946	-0.396456	0.207138	-0.385957	0.226132	-0.378072	0.245939	-0.37048
0.266567	-0.360285	0.288015	-0.347137	0.310274	-0.332313	0.333338	-0.31696
0.357205	-0.301702	0.381869	-0.286996	0.407327	-0.273398	0.433577	-0.26107

0.460619	-0.249829	0.488450	-0.239440	0.517072	-0.229495	0.546486	-0.21868
0.576693	-0.204904	0.607695	-0.186239	0.639487	-0.163666	0.672057	-0.14166
0.705396	-0.124829	0.739503	-0.112534	0.774389	-0.097547	0.810060	-0.07234
0.846512	-0.038530	0.883731	-0.006437	0.921712	0.025749	0.960469	0.07970
1.000000	0.142642						

APPENDIX E

THIS APPENDIX PRESENTS THE SOURCE CODE FOR THE POTENTIAL FLOW PROGRAM FLO27.

```
C********************FLO27********9/12/83***********************************
C**                                                                       **
C**    THREE DIMENSIONAL WING ANALYSIS IN TRANSONIC FLOW USING            **
C**    FINITE VOLUME SCHEME WITH SHEARED PARABOLIC COORDINATES            **
C**    PROGRAMMED BY ANTONY JAMESON, JANUARY-APRIL 1977                   **
C**                                                                       **
C**************************************************************************
C**    THE BOEING VERSION OF FLO27 WAS PREPARED BY DR. HAI-CHOW           **
C**    CHEN WITH THE FOLLOWING MODIFICATIONS                              **
C**       1) TEMPORARY STORAGE OF THE LARGE CORE MEMORY REQUIREMENTS      **
C**          HAS BEEN IMPLEMENTED TO REDUCE THE COMPUTING COSTS           **
C**          BY BUFFERING DATA IN AND OUT OF CORE.                        **
C**                                                                       **
C**       2) STANDARD BOEING INPUT FORMAT FOR THE WING SECTION            **
C**          HAS BEEN USED.                                               **
C**                                                                       **
C**       3) SUBPROGRAM BLIN HAS BEEN IMPLEMENTED TO ADD THE              **
C**          DISPLACEMENT THICKNESS TO THE ORIGINAL WING SECTIONS         **
C**                                                                       **
C**       4) WING SECTION LEADING EDGE SINGULAR POINT IS FOUND BY         **
C**          COMPUTING THE FOCUS OF A PARABOLA BY 2N+1 POINTS             **
C**          LEAST-SQUARE FIT CENTERED AT THE LEADING EDGE POINT.         **
C**          N IS SUPPLIED BY THE USER THROUGH INPUT CARD.                **
C**                                                                       **
C**       5) TRAILING EDGE CLOSURE ANGLE AND BISECTOR SLOPE ARE           **
C**          COMPUTED BASED ON BACKWARD DIFFERENCE.                       **
C**                                                                       **
C**       6) OPTION FOR PRINTER-PLOTTING OF THE UNWRAPPED                 **
C**          WING SECTIONS IS AVAILABLE                                   **
C**                                                                       **
C**************************************************************************
C**************************************************************************
C
C      THE FOLLOWING FILES ARE USED TO EXECUTE FLO27. SOME OF THESE
C      FILES ARE USED SUBSIQUENTLY IN OTHER MODULES OF THE VISCOUS/
C      INVISCID INTERACTIVE WING SYSTEM
C
C          FILE1 IS USED TO BUFFER DATA IN AND OUT OF CORE
C
C          FILE2 IS USED TO BUFFER DATA IN AND OUT OF CORE
C
C          FILE3 IS USED TO BUFFER DATA IN AND OUT OF CORE
```

```
C
C         FILE4 IS USED TO READ IN THE VELOCITY POTENTIAL
C         GENERATED PREVIOUSLY
C
C         FILE8 IS WRITTEN FOR DATA TRANSFER TO BOEING
C         TURBULENT BOUNDARY LAYER PROGRAM A411.
C
C         FILE9 IS USED TO SAVE SECTION SURFACE PRESSURE
C         AGAINST X/C.
C
C         FILE10 IS USED TO SAVE THE SECTION X, Y, Z
C         LOCATION FOR A411IN WHICH CALCULATES THE CORRESPONDING
C         DISPLACEMENT THICKNESS
C         WHERE X, Y, Z SHOULD BE THE WING SURFACE LOCATION
C         FOR THE CURRENT RUN
C
C         FILE11 IS USED TO READ IN THE DISPLACEMENT
C         THICKNESS FROM A411IN
C
C         FILE12 IS USED TO REPLACE PART OF THE INPUT CARDS
C         BY CARD IMAGES
C
C         TAPE13 IS USED TO SAVE PART OF THE OUTPUT
C         SKIPPED FROM THE LINE PRINTER
C
C         TAPE14 IS USED TO SAVE THE VELOCITY POTENTIAL
C         FOR FUTURE USE
C*******************************************************************
C*******************************************************************
      COMMON       G(161,18,3),SO(161,35),VORT(115),ZV(115),
     1             IV(161,35),ITE1(35),ITE2(35),
     2             AO(161),BO(18),XO(35),YO(35),ZO(35),SCAL(35),
     3             NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,FUS,
     4             YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
      COMMON /OPF/ NL
      COMMON /PCKR/  PTCK
      COMMON/FLO/ P1 ,P2,P3,FRES,IRES,JRES,KRES,ARES,CG,IG,JG,KG,AG,NSUP
      COMMON /PARMT3/ XT3(161), YT3(161), ZT3(161),
     1                UT3(161),VT3(161),WT3(161),NOI,Z
      COMMON /PRS/ XOCD(161)
      DIMENSION    XS(161,11),YS(161,11),ZS(11),XLE(11),YLE(11),
     1             SLOFT(11),TRAIL(11),NP(11),E1(11),E2(11),E3(11),
     2             D1(161),D2(161),D3(161),SN(161),
     3             SV(161),SM(161),CP(161),XP(161),YP(161),
     4             YPO(35),ZPO(35),XMAX(35),XMIN(35),YMAX(35),YMIN(35),
     5             CHORD(35),SCL(35),SCD(35),SCM(35),TITLE(20)
      DIMENSION    FIT(3),COVO(3),P10(3),P20(3),P30(3),FSMOD(3),
     1             RES(201),CCUNT(201),PTMAP(3),
```

```
     2                DUMX(161),DUMY(161),DELR(161)
C*****************************************************************************
C** INITIALIZE INPUT PARAMETERS WHICH HAVE RECOMMENDED PROGRAM VALUES **
C**                                                                         **
      XSCAL     = 0.0
      PSCAL     = 0.0
      FCCNT     = 0.0
      BLCP      = 0.0
      WEIG      = 1.0
      PTCK      = 0.0
C**                                                                         **
C*****************************************************************************
      ND        = 161
      NE        = 161
      KPLCT     = 0
      IPLCT     = 1
      ISTCP     = 2
      N1        = 1
      N2        = 2
      N3        = 3
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 4
      REWIND 10
      REWIND 13
      REWIND 14
      JO        = 0
      RAD       = 57.295779513082
    1 WRITE (6,600)
      WRITE (6,2)
    2 FORMAT('CA460 MODIFIED FROM FLO27 OF ANTONY JAMESON,',
     1'COURANT INSTITUTE'/
     1        50H0THREE DIMENSIONAL WING ANALYSIS IN TRANSONIC FLOW,
     2        27H USING FINITE VOLUME SCHEME/
     3 '0BOEING   VERSION, PREPARED BY DR. HAI-CHOW CHEN',
     4 ' STANDARD BOEING   INPUT FORMAT FOR WING SECTION DATA IS USED')
      READ  (5,530) TITLE
      WRITE (6,630) TITLE
      READ  (5,500)
      READ  (5,510)  FNX,FNY,FNZ,FMESH,FPLOT
      NX        = FNX
      NY        = FNY
      NZ        = FNZ
      MMESH     = FMESH
      IF (NX.LT.1) GO TO 301
      KPLOT     = ABS(FPLOT)
      READ  (5,500)
```

```
      DO 12 NM=1,NMESH
C***********************************************************************
C** INITIALIZE INPUT PARAMETERS WHICH HAVE RECOMMENDED PROGRAM VALUES **
C**                                                                   **
      P2C(NM)    = 0.7
      P3C(NM)    = 1.0
      FSMCO(NM)  = 0.0
      PTMAP(NM)  = 0.0
C**                                                                   **
C***********************************************************************
   12 READ  (5,510)  FIT(NM),CCVO(NM),P10(NM)
      READ  (5,500)
      READ  (5,510)  FMACH,YA,AL,CDO
      CALL GEOM   (ND,NS,NP,XS,YS,ZS,XLE,YLE,SLOPT,TRAIL,XP,YP,
     1             FUS,XTEO,CHORDO,ZTIP,SWEEP,DIHED,
     2             FIX,PX,PZ,ISYMO,KSYM)
      ALPHA = AL/RAD
      IF (BLCF.LE.0.) GO TO 44
      IF (PTCK .GE. 1.) WRITE (6,600)
      READ (11) (TITLE(I),I=1,8),FMACH,ALPHA,NS
      DO 40 K = 1,NS
      READ (11) NPCK
      NP(K) = NPCK
      NPCK1 = NPCK + 1
      READ (11) (DUMX(NPCK1-I),DUMZ,DUMY(NPCK1-I),DELR(NPCK1-I),
     1           I=1,NPCK)
      IF (PTCK .LE. 0.) GO TO 30
      WRITE (6,52) K,NPCK,NPCK1
      WRITE (6,54) (DUMX(NPCK1-I),DUMZ,DUMY(NPCK1-I),
     1              DELR(NPCK1-I),I=1,NPCK)
      WRITE (6,54) (XS(I,K),YS(I,K),I=1,NPCK)
   30 CONTINUE
      CALL BLIN (XS(1,K),YS(1,K),DELR,WEIG,NPCK,NL)
   40 CONTINUE
   44 CONTINUE
      IF (PTCK .LT. 0.) GO TO 56
      IF (PTCK .GE. 1.) WRITE (6,600)
      WRITE (10) (TITLE(I),I=1,8),FMACH,ALPHA,NS
      DO 50 K=1,NS
      NPCK = NP(K)
      NPCK1 = NPCK + 1
      DO 48 I = 1,NPCK
      DUMX(NPCK1-I) = XS(I,K) + XLE(K)
      DUMY(NPCK1-I) = YS(I,K) + YLE(K)
   48 CONTINUE
      WRITE (10) NPCK
      WRITE (10) (DUMX(I),ZS(K),DUMY(I),I=1,NPCK)
      IF (PTCK .LE. 0.) GO TO 49
```

```
      WRITE (6,52) K,NPCK,NPCK1,ZS(K)
      WRITE (6,54) (DUMX(I),ZS(K),DUMY(I),I=1,NPCK)
   49 CONTINUE
   50 CONTINUE
      ENDFILE 10
   52 FORMAT (1H0,5X,3I5,F11.4/)
   54 FORMAT (12F11.4)
   56 CONTINUE
      IF (KSYM.NE.0) YA = 0.
      ISYM        = ISYMO
      IF (AL.NE.0.) ISYM = 0
      YAW         = YA/RAD
      CYAW        = COS(YAW)
      SYAW        = SIN(YAW)
      CA          = CYAW*COS(ALPHA)
      SA          = CYAW*SIN(ALPHA)
      IF (FCCNT.LT.1.) GC TC 91
      READ  (4)     NX,NY,NZ,NM,K1,K2,NIT
      MX          = NX   +1
      MY          = NY   +2
      MZ          = NZ   +3
      DO 62 K=1,MZ
      READ  (4)     ((G(I,J,1),I=1,MX),J=1,MY)
C     BUFFER CUT(N3,1) (G(1,1,1),G(MX,MY,1))
      WRITE (N3)  ((G(I,J,1),I=1,MX), J=1,MY)
C     IF (UNIT(N3).GT.0.) GO TO 1
C     BUFFER CUT(N1,1) (G(1,1,1),G(MX,MY,1))
      WRITE (N1)  ((G(I,J,1),I=1,MX),J=1,MY)
C     IF (UNIT(N1).GT.0.) GO TO 1
   62 CONTINUE
      READ  (4)     (VORT(K),K=K1,K2)
      REWIND N3
      REWIND N1
      REWIND 4
   91 CALL CCORD (NX,NY,NZ,KSYM,ZTIP,XLIM,ZLIM,
     1              SY,AX,AY,AZ,PX,PZ,AC,BC,ZO)
      CALL SINGL (NS,NZ,KSYM,KTE1,KTE2,FUS,CHORDO,ZS,XLE,YLE,
     1              SWEEP,DIHED,XO,YO,ZO,YFO,ZFO,E1,E2,E3,IND)
      CALL SUFF  (ND,NE,NS,NX,NZ,ISYM,KSYM,KTE1,KTE2,
     1              YAW,XTEO,XLIM,FIX,NP,XS,YS,ZS,SLOPT,TRAIL,
     2              AO,XO,ZO,SO,SCAL,ZV,IV,ITE1,ITE2,
     3              XP,YP,SN,D1,D2,D3,IND)
      IF (INC.EQ.0) GC TC 291
      IF (FCCNT.GE.1.) GO TO 101
      NM          = 1
      NIT         = 0
      CALL ESTIM
      IF (IO.EQ.0) GO TO 1
```

```
      REWIND N3
      REWIND N1
  101 IF (PTCK .GE. 1.) WRITE (6,600)
      FCCNT        = 0.
      COV          = COVO(NM)
      P1           = P10(NM)
      P2           = P20(NM)
      P3           = P30(NM)
      MIT          = FIT(NM)   +NIT
      KIT          = MIT
      IF (KIT.LT.2) KIT=2
      JIT          = NIT
      LRES         = 0
      MRES         = (MIT  -NIT  -2)/200  +2
      NRES         = 0
      MX           = NX   +1
      MY           = NY   +2
      MZ           = NZ   +3
      KY           = NY   +1
      K1           = 1
      K2           = NZ   +1
      IF (KSYM.EQ.0) GO TO 103
      K1           = 3
      K2           = NZ   +3
  103 LZ           = NZ/2   +1
      IF (KSYM.NE.0) LZ = 3
      IF (PTCK .LE. 0.) GO TO 108
      WRITE (6,104)
  104 FORMAT(48H0INDICATION OF LOCATION OF WING AND VORTEX SHEET,
     1          27H IN COORDINATE PLANE Y = 0./
     2          27H0((IV(I,K),K=K1,K2),I=1,MX))
      DO 106 I=1,MX
  106 WRITE (6,650) (IV(I,K),K=K1,K2)
  108 CONTINUE
      IMAP = FTMAP(NM)
      IF (IMAP .EQ. 0) GO TO 830
      WRITE (6,600)
      WRITE (6,112)
  112 FORMAT(49H0CHORDWISE CELL DISTRIBUTION IN SQUARE ROOT PLANE,
     1          54H AND MAPPED SURFACE COORDINATES AT CENTER LINE AND TIP)
      DO 820 ISEC = LZ,KTE2,IMAP
      WRITE (6,812) ISEC,ZPO(ISEC)
  812 FORMAT (15H0 I        X      ,20H SECTION PROFILE NO.,
     1 I2,3X,'Z =',G13.4)
  820 CALL PPXY (2,NX,AO,SO(1,ISEC))
  830 CONTINUE
      IF (PTCK .LE. 0.) GO TO 130
      WRITE (6,116)
```

```
116 FORMAT(15H0  TE LOCATION ,15H     POWER LAW   )
    WRITE (6,610) XLIM,AX
    WRITE (6,600)
    WRITE (6,118)
118 FORMAT(46H0NORMAL CELL DISTRIBUTION IN SQUARE ROOT PLANE/
   1          15H0         Y       )
    DO 120 J=1,KY
120 WRITE (6,610) BO(J)
    WRITE (6,122)
122 FORMAT(15H0  SCALE FACTOR,15H     POWER LAW   )
    WRITE (6,610) SY,AY
    WRITE (6,600)
    WRITE (6,124)
124 FORMAT(45H0SPANWISE CELL DISTRIBUTICN AND SINGULAR LINE/
   1          15H0         Z      ,15H       X SING   ,15H       Y SING    )
    DO 126 K=K1,K2
126 WRITE (6,610) ZO(K),XO(K),YO(K)
    WRITE (6,128)
128 FORMAT(15H0  TIP LOCATION,15H     POWER LAW   )
    WRITE (6,610) ZLIM,AZ
130 CONTINUE
    WRITE (6,600)
    WRITE (6,132)
132 FORMAT(15H0ITERATIVE SOLUTION)
    WRITE (6,134)
134 FORMAT(15H0     MACH NO    ,15H YAW            ,15H  ANG OF ATTACK)
    WRITE (6,610) FMACH,YA,AL
    WRITE (6,136)
136 FORMAT(15H0     NX         ,15H       NY       ,15H      NZ        )
    WRITE (6,640) NX,NY,NZ
    WRITE (6,138)
138 FORMAT(15H0  RELAX FCT 1 ,15H  RELAX FCT 2  ,15H  RELAX FCT 3   )
    WRITE (6,610) P10(NM),P20(NM),P30(NM)
    WRITE (6,140)
140 FORMAT(10H0ITERATION,
   1          15H   MAX CORRECN ,4H  I ,4H  J ,4H  K ,15H   AVG CORRECN ,
   2          15H   MAX RESIDAL ,4H  I ,4H  J ,4H  K ,15H   AVG RESIDAL ,
   3          12H CIRCULATION,15H      SONIC PTS)
141 NIT          = NIT  +1
    JIT          = JIT  +1
    CALL MIXFLO
    IF (IO.EQ.0) GO TO 151
    JO           = 0
    REWIND N1
    REWIND N2
    N            = N1
    N1           = N2
    N2           = N3
```

```
      N3          = N
      WRITE (6,660) NIT,DG,IG,JG,KG,AG,FRES,IFES,JRES,KRES,ARES,
     1                      VORT(LZ),NSUP
      LRES        = LRES   +1
      IF (LRES.EQ.MRES) LRES = 1
      IF (LRES.NE.1) GO TO 143
      NRES        = NRES   +1
      COUNT(NRES) = NIT   -1
      RES(NRES) = FRES
  143 IF (JIT.GE.KIT) GO TO 251
      IF (NM.LE.1 .OR. NM.LT.MMESH) GO TO 148
      IF (ABS(DG) .LE. COV) GO TC 251
  148 CONTINUE
      IF (NIT.LT.MIT.AND.ABS(DG).GT.COV.AND.ABS(DG).LT.10.) GO TO 141
      GO TO 161
  151 IF (JO.EQ.1) GO TO 1
      REWIND N1
      REWIND N2
      JO          = 1
      N           = N3
      N3          = N2
      N2          = N1
      N1          = N
      GO TO 141
  161 RATE        = 0.
      IF (NRES.GT.1) RATE = (ABS(RES(NRES)/RES(1)))
     1                      **(1./(COUNT(NRES)  -COUNT(1)))
      WRITE (6,162)
  162 FORMAT(15H0 MAX RESIDAL 1,15H  MAX RESIDAL 2,15H         WORK    ,
     1         15H  REDUCTN/CYCLE, 15H CONV TOLERENCE)
      WRITE (6,670) RES(1),RES(NRES),COUNT(NRES),RATE
     1,COV
      WRITE (6,600)
      DO 164 M=1,3
C     BUFFER IN (N1,1) (G(1,1,M),G(MX,MY,M))
      READ (N1,ERR=151) ((G(I,J,M),I=1,MX),J=1,MY)
C     IF (UNIT(N1).GT.0.) GO TO 151
  164 CONTINUE
      LX          = NX/2   +1
      K           = 2
      KKK         = 0
C               WRITE HEADER ON TAPE 8
      IF(NM.LT.MMESH)GO TO 170
      REWIND 8
      REWIND 9
      NRC = KTE2 - KTE1 + 1
      WRITE (8) (TITLE(I),I=1,8),FMACH,ALPHA,NRC
  170 CONTINUE
```

```
171 K             = K   +1
    IF  (K.EQ.MZ) GO TO 191
    DO  172  J=1,MY
    DO  172  I=1,MX
    G(I,J,1)   = G(I,J,2)
172 G(I,J,2)   = G(I,J,3)
C   BUFFER  IN  (N1,1) (G(1,1,3),G(MX,MY,3))
    READ  (N1,ERR=151)  ((G(I,J,3),I=1,MX),J=1,MY)
C   IF  (UNIT(N1).GT.0. ) GO TO 151
    IF  (K.LT.KTE1.OR.K.GT.KTE2) GO TO 171
    CALL  VELO   (K,SV,SM,CP,XP,YP,XMAX(K),XMIN(K),YMAX(K),YMIN(K))
    I1           = ITE1(K)
    I2           = ITE2(K)
    CHORD(K)   = XP(I1)   -XP(LX)
    CALL  FCRCF  (I1,I2,XP,YP,CP,AL,CHORD(K),XO(K),YPO(K),
   1              SCL(K),SCD(K),SCM(K))
    KKK = KKK +1
    IF  (KPLOT.GT.1.AND.K.GT.KTE1)  GO TO 185
    IF  (KPLOT .EQ.  0 .AND.  KKK  .GT.  1)  GO TO 185
    WRITE  (6,600)
    WRITE  (6,182)
182 FORMAT(24HOSECTION  CHARACTERISTICS/
   1        15HO      MACH NO    ,15H YAW               ,15H  ANG OF ATTACK)
    WRITE  (6,610)  FMACH,YA,AL
    WRITE  (6,184)
184 FORMAT(15HOSPAN  STATION   ,15H      CL          ,15H     CD         ,
   1        15H       CM       )
185 WRITE  (6,610)  ZPO(K),SCL(K),SCD(K),SCM(K)
    Z = ZPO(K)
    IF  (MIT.LE.0)  GO TO 850
    IF  (KPLOT.LE.2)  CALL  CPLOT  (2,NX,FMACH,XP,YP,CP,SM,I1,I2,KPLOT)
850 CONTINUE
C              WRITE  ONE  FILE  ON  TAPE 8
    IF(NM.LT.MMESH)GO TO 186
    WRITE  (8)  NCI
    WRITE  (8)  (XT3(I),ZT3(I),YT3(I),UT3(I),WT3(I),VT3(I),I=1,NCI)
    NRC=I2-I1+1
C           WRITE  CP  VS  X/C  SECTION  DATA  FOR  FINAL  MESH  ON  TAPE  9
    WRITE  (9,900)  ZPO(K)
    WRITE  (9,910)  NRD
    WRITE  (9,920)
    WRITE  (9,950)  (XOCD(J),CP(J),J=I1,I2)
900 FORMAT  (/,14HSPAN  STATION =,F12.5)
910 FORMAT  (1X,20HNO.  OF  DATA  POINTS = ,I5)
920 FORMAT  (1X,6H    X/C,8X,2HCP,7X,3HX/C,8X,2HCP,7X,3HX/C,8X,2HCP,
   17X,3HX/C,8X,2HCP,3X)
950 FORMAT  (8F10.6)
186 CONTINUE
```

```
C
C WHEN KPLOT = 2 CALL SUBROUTINE VERTEC WHICH PLOTS CP VS X/C
C FOR EACH SECTION OF THE FINAL MESH
C
      IF (KPLOT.EQ.2.AND.NM.EQ.MMESH) CALL VERTEC(I1,I2,XOCD,CP,NRD,
     1                                ZPO,FMACH,YA,AL,SCL,SCD,SCM,K)
      GO TO 171
  191 CONTINUE
      IF(NM.LT.MMESH)GO TO 200
      ENDFILE 8
      REWIND 8
      ENDFILE 9
      REWIND 9
  200 CONTINUE
      CALL TOTFOR (KTE1,KTE2,CHORD,SCL,SCD,SCM,XC,YPO,ZPO,
     1            CL,CD1,CMP,CMR,CMY)
      CD1         = CYAW*CD1
      CD          = CD0  +CD1
      VLD1        = 0.
      IF (ABS(CD1).GT.1.E-6) VLD1 = CL/CD1
      VLD         = 0.
      IF (ABS(CD).GT.1.E-6) VLD = CL/CD
      WRITE (6,600)
      WRITE (6,192)
  192 FORMAT(21HOWING CHARACTERISTICS/
     1       15H0     MACH NO    ,15H YAW              ,15H  ANG OF ATTACK)
      WRITE (6,610) FMACH,YA,AL
      WRITE (6,194)
  194 FORMAT(15H0        CL      ,15H  CD FORM       ,15H  CD FRICTION  ,
     1       15H       CD        ,15H   L/D FORM     ,15H  L/D          )
      WRITE (6,610) CL,CD1,CD0,CD,VLD1,VLD
      WRITE (6,196)
  196 FORMAT(15H0      CM YAW    ,15H  CM ROLL       ,15H  CM PITCH     )
      WRITE (6,610) CMY,CMR,CMP
      REWIND N1
      IF (KPLOT.LT.1) GO TO 201
C     CALL RPLOT (IPLOT,NRES,RES,COUNT,TITLE,FMACH,YA,AL,NX,NY,NZ)
C     CALL DRAW (IPLOT,XMAX,XMIN,YMAX,YMIN,ZPO,FUS,TITLE,NZ,KTE1,KTE2)
C     CALL THREED (IPLOT,SV,SM,CP,XP,YP,ZPO,TITLE,YA,AL,
C    1             VLD,CL,CD,CHORDO,XSCAL,PSCAL)
      IF (IO.EQ.0) GO TO 151
  201 IF (ISTOP.EQ.1) GO TO 301
      IF (NM.LT.MMESH) GO TO 203
      GO TO 1
  203 CONTINUE
      NX          = NX   +NX
      NY          = NY   +NY
      NZ          = NZ   +NZ
```

```
      CALL CCCRD (NX,NY,NZ,KSYM,ZTIP,XLIM,ZLIM,
     1            SY,AX,AY,AZ,PX,PZ,AO,BC,ZO)
      CALL SINGL (NS,NZ,KSYM,KTE1,KTE2,FUS,CHCRDO,ZS,XLE,YLE,
     1            SWEEP,DIHED,XO,YO,ZO,YFC,ZPO,E1,E2,E3,IND)
      CALL SUFF  (ND,NE,NS,NX,NZ,ISYM,KSYM,KTE1,KTE2,
     1            YAW,XTEO,XLIM,FIX,NP,XS,YS,ZS,SLOPT,TRAIL,
     2            AO,XO,ZO,SO,SCAL,ZV,IV,ITE1,ITE2,
     3            XP,YP,SN,C1,D2,D3,IND)
      IF (INC.EQ.0) GC TC 291
      CALL REFIN
      IF (IO.EQ.0) GO TO 221
      REWIND N1
      REWIND N2
      NSMOO        = FSMOO(NM)
      IF (NSMOO.LT.1) GO TO 211
      DO 202 N=1,NSMOO
      CALL SMOO
      IF (IO.EQ.0) GO TO 221
      REWIND N1
  202 REWIND N2
  211 N            = N1
      N1           = N2
      N2           = N3
      N3           = N
      NM           = NM   +1
      NIT          = 0
      GO TO 101
  221 NX           = NX/2
      NY           = NY/2
      NZ           = NZ/2
      CALL CCCRD (NX,NY,NZ,KSYM,ZTIP,XLIM,ZLIM,
     1            SY,AX,AY,AZ,PX,PZ,AO,BC,ZO)
      CALL SINGL (NS,NZ,KSYM,KTE1,KTE2,FUS,CHCRDO,ZS,XLE,YLE,
     1            SWEEP,DIHED,XO,YO,ZO,YFC,ZPC,E1,E2,E3,IND)
      CALL SUFF  (ND,NE,NS,NX,NZ,ISYM,KSYM,KTE1,KTE2,
     1            YAW,XTEO,XLIM,FIX,NP,XS,YS,ZS,SLOPT,TRAIL,
     2            AO,XO,ZO,SO,SCAL,ZV,IV,ITE1,ITE2,
     3            XP,YP,SN,D1,D2,D3,IND)
      IF (INC.EQ.0) GC TO 291
      GO TO 151
  251 K1           = KTE1   -1
      K2           = KTE2   +ITE2(KTE2)   -NX/2
      WRITE (14) NX,NY,NZ,NM,K1,K2,NIT
      DO 262 K=1,NZ
C     BUFFER IN (N1,1) (G(1,1,1),G(MX,MY,1))
      READ (N1,ERR=281) ((G(I,J,1),I=1,MX),J=1,MY)
C     IF (UNIT(N1).GT.0.) GO TO 281
  262 WRITE (14) ((G(I,J,1),I=1,MX),J=1,MY)
```

```
      REWIND N1
      WRITE (14) (VORT(K),K=K1,K2)
      ENDFILE 14
      REWIND 14
C     CALL SSWTCH(1,ISTOP)
      CALL SLITET(1,ISTOP)
      IF (ISTOP.EQ.1) GO TO 161
      JIT      = 0
      IF (NIT.LT.MIT.AND.ABS(DG).GT.COV.AND.ABS(DG).LT.10.) GO TO 141
      GO TO 161
  281 REWIND 4
      GO TO 151
  291 WRITE (6,600)
      WRITE (6,292)
  292 FORMAT(24H0BAD DATA,SPLINE FAILURE)
      GO TO 1
  301 CONTINUE
      REWIND 10
      REWIND 11
C     IF (KPLOT.GT.0)  CALL PLOT(0.,0.,999)
      STOP
  500 FORMAT(1X)
  510 FORMAT(8F10.6)
  530 FORMAT(20A4)
  600 FORMAT(1H1)
  610 FORMAT(F12.5,7G15.5)
  630 FORMAT(1H0,20A4)
  640 FORMAT(I8,7I15)
  650 FORMAT(1X,33I3)
  660 FORMAT(I10,E15.5,3I4,2E15.5,3I4,E15.5,F10.5,I10,F10.3)
  670 FORMAT (2E15.4,2F15.4,E15.4)
      END
```

```
C**SUBROUTINE BLIN*************************************************************
      SUBROUTINE BLIN (XT,YT,DELR,WEIG,N,NL)
C
C     SUBPROGRAM FOR NORMALLY ADDING THE DISPLACEMENT
C     THICKNESS TO THE ORIGINAL WING SECTIONS
C
C     XT:      CONTAIN THE X COORDINATES OF THE ORIGINAL
C              WING SECTION WHEN CALLED
C              CONTAIN THE X COORDINATES OF THE DISPLACED
C              WING SECTION ON RETURN
C     YT:      CONTAIN THE Y COORDINATES OF THE ORIGINAL
C              WING SECTION WHEN CALLED
C              CONTAIN THE Y COORDINATES OF THE DISPLACED
C              WING SECTION ON RETURN
C
C     DELR:    THE DISPLACEMENT THICKNESS
C
      COMMON /BCKR/  PTCK
      DIMENSION XT(1),YT(1),DELR(1)
      WRITE (6,1000)
      I = 1
      X2 = XT(I)
      Y2 = YT(I)
  200 IF (I.EQ.N) GO TO 300
      X3 = XT(I+1)
      Y3 = YT(I+1)
  300 IF (I.NE.1) GO TO 400
      X1 = 2.*X2 - X3
      Y1 = 2.*Y2 - Y3
  400 IF (I.NE.N) GO TO 500
      X3 = 2.*X2 - X1
      Y3 = 2.*Y2 - Y1
  500 CONTINUE
      X12 = X1 - X2
      X23 = X2 - X3
      X31 = X3 - X1
      IF (ABS(X31).LE.1.E-6) GO TO 600
      DYX = -X23/(X12*X31)*Y1
      DYX = DYX + (X12-X23)/(X12*X23)*Y2
      DYX = DYX + X12/(X31*X23)*Y3
      IF (ABS(DYX).LE.1.E-6) GO TO 820
      DYXN =-1./DYX
      GO TO 700
  600 DYXN = 0.
  700 CONTINUE
      IF (I.EQ.NL) GO TO 880
  800 CONTINUE
      SI = 1.
```

```
      IF (I.LT.NL) SI = -1.
      GO TO 850
  820 CONTINUE
      S = 0.
      DL = DELR(I)
      DX = 0.
      DY = DL*SI
      GO TO 880
  850 CONTINUE
      S = 1. + DYXN**2
      S = SQRT(S)
      S = 1./S
      F = 1.
      IF (DYXN *SI.LT.0.) F = -1.
      DL = DELR(I)
      DX = S*F
      DY = ABS(DYXN)*S*SI
      DX = DX * DL
      DY = DY * DL
      XT(I) = X2 + DX*WEIG
      YT(I) = Y2 + DY*WEIG
  880 CONTINUE
      IF (PTCK .LE. 1.) GO TO 890
      WRITE (6,1000) I,F,S,DL,DX,DY,DYXN,XT(I),YT(I),DYX,WEIG
  890 CONTINUE
      IF (I.EQ.N) GO TO 900
      X1 = X2
      Y1 = Y2
      X2 = X3
      Y2 = Y3
      I = I+1
      GO TO 200
  900 CONTINUE
      RETURN
 1000 FORMAT (1H ,I5,F7.2,9G13.5)
      END
```

```
C**SUBROUTINE PPXY***********************************************************
      SUBROUTINE PPXY (I1,I2,X,Y)
C
C     SUBPROGAM FOR LINE PRINTER PLOTTING OF THE UNWRAPPED
C     WING SECTIONS
C
      COMMON /PCKR/ PTCK
      COMMON /SHARE/ LINE(100)
      DIMENSION X(1),Y(1)
      DATA IB /1H /, IP /1H+/, KMAX /100/, ADD /1.5/,
     1 IZ /1H./, ICONST /0/
      DO 10 I=1,100
      LINE(I) = IB
   10 CONTINUE
      YMAX = -1.0E35
      YMIN = -YMAX
      WIDTH = KMAX - 5
      DO 20 I = I1,I2
      YMAX = AMAX1(YMAX,Y(I))
      YMIN = AMIN1(YMIN,Y(I))
   20 CONTINUE
      VAL = ABS(YMAX) + ABS(YMIN)
      S = WIDTH/VAL
      KK = 0
      IF (ICONST.LE.YMAX .AND. ICONST.GE.YMIN) KK = S*(YMAX-ICONST)+ADD
      IF (KK.NE.0) LINE(KK)=IZ
      DO 30 I=I1,I2
      K = S * (YMAX-Y(I)) + ADD
      IF (K.LT.1) K = 1
      IF (K.GT.KMAX) K = KMAX
      LINE(K) = IP
      WRITE (6,100) I,X(I),Y(I),LINE
      LINE(K) = IB
      IF (K.EQ.KK) LINE(KK) = IZ
   30 CONTINUE
      RETURN
  100 FORMAT (1X,I3,2F10.4,4X,100A1)
      END
```

```
C**SUBROUTINE LSQR*************************************************************
      SUBROUTINE LSQR (NL,NB,XP,YP,XSING,YSING)
C
C     SUBPROGAM FOR WING SECTION LEADING EDGE SINGULAR POINT
C     CALCULATION BY MEANS OF COMPUTING THE FOCUS OF A
C     PARABOLA BY NB*2+1 POINTS LEAST-SQUARE FIT CENTERED AT
C     THE LEADING EDGE
C
C     NB :      SUPPLY BY THE CALLING PROGRAM GEOM
C
      COMMON /PCKR/ FTCK
      DIMENSION XP(1),YP(1)
      N1 = NL - NB
      N2 = NL + NB
      N = N2 - N1 + 1
      A1 = N
      B1 = 0.
      C1 = 0.
      A2 = 0.
      B2 = 0.
      C2 = 0.
      A3 = 0.
      B3 = 0.
      C3 = 0.
      D1 = 0.
      D2 = 0.
      D3 = 0.
      SCALE = 100.
      SCALE2 = 500.
      DO 300 I = N1,N2
      YY =(YP(I) - YP(NL))*SCALE
      B1 = B1 + YY
      YP2 = YY*YY
      C1 = C1 + YP2
      YP3 = YP2 * YY
      C2 = C2 + YP3
      YP4 = YP3 * YY
      C3 = C3 + YP4
      XX = XP(I)*SCALE2
      D1 = D1 + XX
      YX = YY    * XX
      D2 = D2 + YX
      Y2X = YP2 * XX
  300 D3 = D3 + Y2X
      A2 = B1
      B2 = C1
      A3 = C1
      B3 = C2
```

```
      FA1 = B2*C3 - B3*C2
      FA2 = A3*C2 - A2*C3
      FA3 = A2*B3 -A3*B2
      DET = A1*FA1 + B1*FA2 + C1 *FA3
      DI = 1./DET
      FA1 = FA1*DI
      FA2 = FA2*DI
      FA3 = FA3*DI
      FB1 = (B3*C1 - B1*C3)*DI
      FB2 =(A1*C3 - A3*C1) *DI
      FB3 = (A3*B1 - A1*B3)*DI
      FC1 = (B1*C2 - B2*C1)*DI
      FC2 = (A2*C1 - A1*C2)*DI
      FC3 = (A1*B2 - A2*B1)*DI
      A = FA1*D1 + FB1*D2 + FC1*D3
      B = FA2 *D1 +FB2*D2 + FC2*D3
      C = FA3 *D1 +FB3*D2 + FC3*D3
      B = B*SCALE
      C = C*SCALE**2
      A = A/SCALE2
      B = B/SCALE2
      C = C/SCALE2
      XSING = A + (1. - B**2)*.25/C
      YSING = -B*.5/C + YP(NL)
      IF (PTCK .LE. 0.) GO TO 520
      WRITE (6,400) N
  400 FORMAT (1H0,5X,'LEAST SQUARE MATRIX FOR LEADING EDGE',
     1 I3,' POINTS LEAST-SQUARE FIT'/)
      WRITE (6,500) A1,B1,C1,D1
      WRITE (6,500) A2,B2,C2,D2
      WRITE (6,500) A3,B3,C3,D3
      WRITE (6,500) FA1,FB1,FC1
      WRITE (6,500) FA2,FB2,FC2
      WRITE (6,500) FA3,FB3,FC3
      WRITE (6,500) XSING,YSING,A,B,C,DET,DI
  500 FORMAT (10G13.5)
  520 CONTINUE
      R2=0.
      DXAM = 0.
      DX2M = 0.
      DO 650 I = N1,N2
      Y = YP(I) - YP(NL)
      X = A + B*Y + C*Y**2
      DX = X - XP(I)
      DX2 = DX*DX
      R2 = R2 + DX2
      DXA = ABS(DX)
      IF (DXAM.GE.DXA) GO TO 600
```

```
      DXAM = DXA
      DXAM = DXA
      DX2M = DXAM**2
  600 CONTINUE
      IF (PTCK .LE. 0.) GO TO 650
      WRITE (6,700) I,X,Y,DX,DX2,R2,DXAM,DX2M,XP(I),YP(I)
  650 CONTINUE
      IF (PTCK .LE. 0.) GO TO 750
      RA = R2/A1
      WRITE (6,700) N,RA,DXAM
  700 FORMAT (I13,9G13.5)
  750 CONTINUE
      RETURN
      ENTRY LSQ
      IF (DXAM.LE.1.E-4) RETURN
      WRITE (6,800) DXAM
  800 FORMAT (1H0,5X,'WARNING ??? DEVIATION OF THE LEADING EDGE POINTS',
     1 ' FROM PARABOLA IS GREATER THAN 0.0001'/6X,'DXAM =',G13.4/)
      RETURN
      END
```

```
C**SUBROUTINE GEOM*************************************************************
      SUBROUTINE  GEOM  (ND,NS,NP,XS,YS,ZS,XLE,YLE,SLOPT,TRAIL,XP,YP,
     1                   FUS,XTEO,CHORDO,ZTIP,SWEEP,DIHED,
     2                   FIX,PX,PZ,ISYMO,KSYM)
C
C     GEOMETRIC DEFINITION OF WING
C
C     STANDARD BOEING INPUT FORMAT FOR WING SECTION DATA IS USED
C
C     OPTION FOR WING SECTION TRAILING EDGE CLOSURE ANGLE
C     AND BISECTOR SLOP BE AUTOMATIC COMPUTED IS AVAILABLE
C
C     LEADING EDGE SINGULAR POINT CAN BE AUTOMATIC COMPUTED
C     BY INVOKING THE OPTION TO CALL LSQR
C
      COMMON /PCKR/ PTCK
      COMMON /EPF/ NL
      DIMENSION    XS(ND,1),YS(ND,1),ZS(1),XLE(1),YLE(1),
     1             SLOPT(1),TRAIL(1),XP(ND),YP(ND),NP(1)
C******************************************************************************
C**  INITIALIZE INPUT PARAMETERS WHICH HAVE RECOMMENDED PROGRAM VALUES **
C**                                                                    **
      FNB       = 2.0
      PX        = 0.0
      PZ        = 0.0
      FIX       = 0.0
      TRL       = 0.0
      SLT       = 0.0
      XSING     = 0.0
      YSING     = 0.0
      YSYM      = 0.0
C**                                                                    **
C******************************************************************************
      RAD       = 57.295779513082
      READ   (5,500)
      READ   (5,510)  ZSYM,FNS,SWEEP,DIHED,FUS
      IF  (FNS.LT.3.)  RETURN
      KSYM      = ZSYM
      IF  (FUS.GE.0.)  KSYM = 1
      NS        = FNS
      WRITE  (6,2)
    2 FORMAT(15HO FUSELAGE RAD )
      WRITE  (6,610)  FUS
      WRITE  (6,4)
    4 FORMAT(15HO       SWEEP    ,15H       DIHED     )
      WRITE  (6,610)  SWEEP,DIHED
      SWEEP     = SWEEP/RAD
      DIHED     = DIHED/RAD
```

```
      ISYMO      = 1
      XTEC       = 0.
      CHORDO     = 0.
      K          = 1
   11 READ  (5,500)
      READ  (5,510)  ZS(K),XL,YL,CHORD,THICK,AL,FSEC
      IF  (K.EQ.1) ZS1 = ZS(1)
      ALPHA       = AL/RAD
      IF  (K.GT.1.AND.FSEC.EQ.0.) GO TO 31
      READ (5,500)
      READ (5,510) FN
       N = FN
      READ  (5,500)
      N1 = N + 1
      READ (5,520) (XP(N1-I),YP(N1-I),I=1,N)
  520 FORMAT (6F10.0)
      DO 26 I=1,N
      IF  (XP(I+1).LT.XP(I)) GO TO 26
      NL = I
      GO TO 800
   26 CONTINUE
  800 CONTINUE
      IF  (FNB .LE. 0.) GO TO 21
      DYL = YP(1) - YP(2)
      DXL = XP(1) - XP(2)
      DYU = YP(N) - YP(N-1)
      DXU = XP(N) - XP(N-1)
      TSU = DYU/DXU
      TSL = DYL/DXL
      TRL = ATAN2(DYL,DXL) - ATAN2(DYU,DXU)
      TRL = TRL*RAD
      SLT = (TSL+TSU)*.5
      NB = FNB
      CALL LSQR (NL,NB,XP,YP,XSING,YSING)
   21 WRITE (6,600)
      WRITE (6,22) ZS(K)
   22 FORMAT(16H0PROFILE AT Z = ,F10.5/
     1        15H0    TE ANGLE   ,15H     TE SLOPE   ,15H      X SING    ,
     2        15H        Y SING    )
      WRITE (6,610) TRL,SLT,XSING,YSING
   27 WRITE (6,620) NL,XP(NL),(XP(I),YP(I),I=1,N)
  620 FORMAT (///' NL = ' ,I3,',  XP(NL) = ' ,F10.5 //' (XP,YP)'//
     1 (2X,5(2F10.5,4X)))
   31 SCALE      = CHORD/(XP(1)  -XP(NL))
      XLE(K)     = XL   +(XSING  -XP(NL))*THICK*SCALE
      YLE(K)     = YL   +(YSING  -YP(NL))*THICK*SCALE
      XX         = XP(NL)  +(XSING  -XP(NL))*THICK
      YY         = YP(NL)  +(YSING  -YP(NL))*THICK
```

```
      CA          = COS(ALPHA)
      SA          = SIN(ALFHA)
      DO 32 I=1,N
      XS(I,K)     = SCALE*((XP(I)  -XX)*CA  +THICK*(YP(I)  -YY)*SA)
   32 YS(I,K)     = SCALE*(THICK*(YP(I)  -YY)*CA  -(XP(I)  -XX)*SA)
      SLOPT(K)    = THICK*SLT  -TAN(ALPHA)
      TRAIL(K)    = THICK*TRL/RAD
      NP(K)       = N
      CHORDO      = AMAX1(CHORDO,CHORD)
      IF (YSYM.LE.O..OR.ALPHA.NE.O.) ISYMC = C
      WRITE (6,42) ZS(K)
   42 FORMAT(27HOSECTION DEFINITION AT Z = ,F10.5/
     1        15HO      XLE     ,15H      YLE      ,15H      CHORD     ,
     2        15HTHICKNESS RATIO,15H      TWIST     )
      WRITE (6,610) XL,YL,CHCRD,THICK,AL
      YMIN = YP(NL)
      YMAX = YMIN
      DO 44 I = 1,N
      IF (YP(I) .GE. YMIN) GO TO 43
      JMIN = I
      YMIN = YP(I)
   43 IF (YP(I) .LE. YMAX) GO TO 44
      JMAX = I
      YMAX = YP(I)
   44 CONTINUE
      YDIF = YMAX - YMIN
      NN = N   - 1
      SUM = C.
      DO 46 I = NL,NN
      SUM = SUM + .5*(YP(I)+YP(I+1))*(XP(I+1)-XP(I))
   46 CONTINUE
      NM = NL - 1
      DO 48 I = 1,NM
      SUM = SUM + .5*(YP(I)+YP(I+1))*(XP(I+1)-XP(I))
   48 CONTINUE
      WRITE (6,300)
  300 FORMAT (15HO     YMIN     ,15H     JMIN     ,15H     YMAX      ,
     1 15H     JMAX     ,15H     YDIF     ,15H     AREA     )
      WRITE (6,320) YMIN,JMIN,YMAX,JMAX,YDIF,SUM
  320 FORMAT (1H ,G12.4,I11,G19.4,I11,G19.4,G15.4)
      CALL LSC
      IF (FUS.LE.O.) GO TO 61
      R = AMAX1(O.,(FUS**2 - YLE(1)**2))
      Z = ZS(K) - ZS1 + SQRT(R)
      R           = FUS**2/(YLE(K)**2  +Z**2)
      ZS(K)       = Z*(1.   -R)
      YLE(K)      = YLE(K)*(1.   +R)
      S           = R*XS(NL,K)
```

```
      XLE(K)      = XLE(K)   -S
      DO 52  I=1,N
      XS(I,K)     = XS(I,K)  +S
   52 YS(I,K)     = YS(I,K)*(1.  +R)
   61 K           = K   +1
      IF  (K.LE.NS) GO TO 11
      ZO          = .5*(ZS(1)   +ZS(NS))
      IF  (KSYM.NE.0)  ZO = ZS(1)
      DO 62  K=1,NS
      XTEO        = AMAX1(XTEO,XS(1,K))
   62 ZS(K)       = ZS(K)   -ZO
      ZTIP        = ZS(NS)
      RETURN
  500 FORMAT(1X)
  510 FORMAT(8F10.6)
  600 FORMAT(1H1)
  610 FORMAT(F12.4,7F15.4)
      END
```

```
C**SUBROUTINE COORD*****************************************************
      SUBROUTINE  COORD (NX,NY,NZ,KSYM,ZTIP,XLIM,ZLIM,
     1                   SY,AX,AY,AZ,PX,PZ,AO,BO,ZO)
C     SETS UP STRETCHED PARABOLIC AND SPANWISE COORDINATES
      DIMENSION    AO(1),BO(1),ZO(1)
      PI          = 3.1415926535898
      BOUND       = .95
      AX          = .5
      AY          = .5
      AZ          = .5
      XLIM        = .625*BOUND
      ZLIM        = .625*BOUND
      SY          = .5
      SCALZ       = ZTIP/(1.000001*ZLIM)
      LX          = NX/2  +1
      MX          = NX   +1
      DX          = 2.*BOUND/NX
      Q           = PI/XLIM
      R           = PX/Q
      E           = 1./(1.   +R*SIN(Q)   -XLIM)
      DO 12  I=1,MX
      D           = (I   -LX)*DX
      D           = D   +R*SIN(Q*D)
      IF (ABS(D).LE.XLIM) GO TO 12
      B           = 1.
      IF (D.LT.0.) B  = -1.
      A           = 1.   -((D   -B*XLIM)*E)**2
      C           = A**AX
      D           = B*XLIM   +(D   -B*XLIM)/C
   12 AO(I)       = D
      KY          = NY   +1
      DY          = BOUND/NY
      DO 22  J=1,KY
      D           = (KY   -J)*DY
      A           = 1.   -D*D
      C           = A**AY
   22 BO(J)       = SY*D/C
      LZ          = NZ/2   +1
      K1          = 1
      K2          = NZ   +1
      DZ          = 2.*BOUND/NZ
      Q           = PI/ZLIM
      R           = PZ/Q
      IF (KSYM.EQ.0) GO TO 31
      LZ          = 3
      K1          = 2
      K2          = NZ   +3
      DZ          = BOUND/NZ
```

```
      Q           = Q   +Q
      R           = -PZ/Q
   31 E           = 1./(1.   +R*SIN(Q)   -ZLIM)
      DO 32  K=K1,K2
      D           = (K   -LZ)*DZ
      D           = D   +R*SIN(Q*D)
      IF (ABS(C).LE.ZLIM) GC TO 32
      B           = 1.
      IF (D.LT.O.) B  = -1.
      A           = 1.   -((C   -B*ZLIM)*E)**2
      C           = A**AZ
      D           = B*ZLIM   +(D   -B*ZLIM)/C
   32 ZO(K)       = SCALZ*C
      RETURN
      ENC
```

```
C**SUBROUTINE SINGL*************************************************************
      SUBROUTINE  SINGL (NS,NZ,KSYM,KTE1,KTE2,FUS,CHORDO,ZS,XLE,YLE,
     1                  SWEEP,DIHED,XO,YO,ZO,YPO,ZPO,E1,E2,E3,IND)
C     GENERATES SINGULAR LINE FOR SQUARE ROOT TRANSFORMATION
      DIMENSION    ZS(1),XLE(1),YLE(1),XO(1),YO(1),ZO(1),YPO(1),ZPO(1),
     1             E1(1),E2(1),E3(1)
      K1          = 1
      K2          = NZ   +1
      IF  (KSYM.EQ.0)  GO  TO 11
      K1          = 2
      K2          = NZ   +3
      KTE1        = 3
   11 DO  12  K=K1,K2
      IF  (ZO(K).LT.ZS(1))  KTE1 = K   +1
      IF  (ZO(K).LE.ZS(NS))  KTE2 = K
   12 CONTINUE
      CALL  SFLIF  (1,NS,ZS,XLE,E1,E2,E3,2,0.,2,0.,0,0.,IND)
      CALL  INTPL  (KTE1,KTE2,ZO,XO,1,NS,ZS,XLE,E1,E2,E3,0)
      S           = CHORDO*TAN(SWEEP)
      S1          = CHORDO*E1(1)
      S2          = CHORDO*E1(NS)
      CALL  SFLIF  (1,NS,ZS,YLE,E1,E2,E3,2,0.,2,0.,0,0.,IND)
      CALL  INTPL  (KTE1,KTE2,ZO,YO,1,NS,ZS,YLE,E1,E2,E3,0)
      T           = CHORDO*TAN(DIHED)
      T1          = CHORDO*E1(1)
      T2          = CHORDO*E1(NS)
      XO(KTE1-1)  =  XO(KTE1)  +XO(KTE1)   -XO(KTE1+1)
      YO(KTE1-1)  =  YO(KTE1)  +YO(KTE1)   -YO(KTE1+1)
      IF  (KSYM.NE.0)  GO  TO  31
      N           = KTE1   -1
      DO  22  K=K1,N
      ZZ          = (ZO(K)   -ZO(KTE1))/CHORDO
      A           = EXP(ZZ)
      XO(K)       = XO(KTE1)   +S*ZZ   -(S1   -S)*(1.   -A)
   22 YO(K)       = YO(KTE1)   +T*ZZ   -(T1   -T)*(1.   -A)
   31 N           = KTE2   +1
      DO  32  K=N,K2
      ZZ          = (ZO(K)   -ZO(KTE2))/CHORDO
      A           = EXP(-ZZ)
      XO(K)       = XO(KTE2)   +S*ZZ   +(S2   -S)*(1.   -A)
   32 YO(K)       = YO(KTE2)   +T*ZZ   +(T2   -T)*(1.   -A)
      DO  42  K=K1,K2
      YPO(K)      = YO(K)
      ZPO(K)      = ZO(K)
      IF  (FUS.LE.0.)  GO  TO  42
      A           = .25*(ZO(K)**2   -YO(K)**2)   +FUS**2
      B           = .5*ZO(K)*YO(K)
      S           = SQRT(A**2   +B**2)
```

```
      T            = 0.
      IF (S.GT.0.) T = .5*ATAN2(B,A)
      S            = SQRT(S)
      YPO(K)       = .5*YO(K)   +S*SIN(T)
      ZPO(K)       = .5*ZO(K)   +S*COS(T)
   42 CONTINUE
      RETURN
      END
```

```
C**SUBROUTINE SURF**********************************************************
      SUBROUTINE  SURF  (ND,NE,NS,NX,NZ,ISYM,KSYM,KTE1,KTE2,
     1                   YAW,XTEO,XLIM,FIX,NP,XS,YS,ZS,SLCPT,TRAIL,
     2                   AO,XO,ZO,SO,SCAL,ZV,IV,ITE1,ITE2,
     3                   XP,YP,SN,D1,D2,D3,IND)
C     INTERPOLATES MAPPED WING SURFACE AT MESH POINTS
      DIMENSION   SO(NE,1),XS(ND,1),YS(ND,1),ZS(1),SLOPT(1),TRAIL(1),
     1            AO(1),XO(1),ZO(1),SCAL(1),ZV(1),
     2            XP(1),YP(1),SN(1),D1(1),D2(1),D3(1),
     3            IV(NE,1),NP(1),ITE1(1),ITE2(1)
      PI         = 3.14159265358 98
      TYAW       = TAN(YAW)
      SSO        = XTEO/XLIM**2
      DX         = 2./NX
      LX         = NX/2   +1
      MX         = NX   +1
      MZ         = NZ   +3
      IVO        = 1   -ISYM   -ISYM   -ISYM
      IV1        = -1   -ISYM
      DO 2 K=1,MZ
      ITE1(K)    = MX
      ITE2(K)    = MX
      DO 2 I=1,MX
      IV(I,K)    = -2
    2 SO(I,K)    = 0.
      K          = KTE1
      K2         = 1
   21 K2         = K2   +1
      K1         = K2   -1
      R2         = 1.
      IF  (ZS(K2)   -ZO(K)) 21,25,23
   23 R2         = (ZO(K)   -ZS(K1))/(ZS(K2)   -ZS(K1))
   25 R1         = 1.   -R2
      I2         = (13*NX)/16   +1
      I1         = NX   +2   -I2
      IF  (FIX.EQ.C.) GO TO 31
      C          = R1*XS(1,K1)   +R2*XS(1,K2)
      CC         = SQRT(C/SSO)
      DO 26 I=2,NX
      IF  ((AO(I)   +.5*DX).LT.-CC) I1 = I   +1
      IF  ((AO(I)   -.5*DX).LT.CC) I2 = I
   26 CONTINUE
   31 KK         = K1
      RR         = R1
   41 N          = NP(KK)
      ANGL       = PI   +PI
      U          = 1.
      V          = 0.
```

```
      DO 42 I=1,N
      R            = SQRT(XS(I,KK)**2   +YS(I,KK)**2)
      IF (R.EQ.0.) GO TO 43
      ANGL         = ANGL   +ATAN2((U*YS(I,KK)   -V*XS(I,KK)),
     1                              (U*XS(I,KK)   +V*YS(I,KK)))
      U            = XS(I,KK)
      V            = YS(I,KK)
      R            = SQRT(R   +R)
      XP(I)        = R*COS(.5*ANGL)
      YP(I)        = R*SIN(.5*ANGL)
      GO TO 42
   43 ANGL         = PI
      U            = -1.
      V            = 0.
      XP(I)        = 0.
      YP(I)        = 0.
   42 CONTINUE
      S            = AO(I2)/AMIN1(ABS(XP(1)),ABS(XP(N)))
      SS           = .5/S**2
      DO 44 I=1,N
      XP(I)        = S*XP(I)
   44 YP(I)        = S*YP(I)
      ANGL         = ATAN(SLOPT(KK))
      ANGL1        = ATAN(YS(1,KK)/XS(1,KK))
      ANGL2        = ATAN(YS(N,KK)/XS(N,KK))
      ANGL1        = ANGL   -.5*(ANGL1   -TRAIL(KK))
      ANGL2        = ANGL   -.5*(ANGL2   +TRAIL(KK))
      T1           = TAN(ANGL1)
      T2           = TAN(ANGL2)
      CALL SPLIF (1,N,XP,YP,D1,D2,D3,1,T1,1,T2,0,0.,IND)
      IF (IND.EQ.0) WRITE (6,500) KK,K1,K2,N,RR,R1,R2,ZS(KK)
  500 FORMAT (12H0BAD MAPPING,4I10,4G13.4/)
      CALL INTPL (I1,I2,AO,SA,1,N,XP,YP,D1,D2,D3,0)
      X1           = .25*XS(1,KK)
      A            = SLOPT(KK)*(XS(1,KK)   -X1)
      B            = 1./(XS(1,KK)   -X1)
      ANGL         = PI   +PI
      U            = 1.
      V            = 0.
      M            = I1   -1
      DO 52 I=2,M
      XX           = SS*AO(I)**2
      D            = B*(XX   -X1)
      YY           = YS(1,KK)   +A*ALOG(D)/D
      R            = SQRT(XX**2   +YY**2)
      ANGL         = ANGL   +ATAN2((U*YY   -V*XX),(U*XX   +V*YY))
      U            = XX
      V            = YY
```

```
      R          = S*SQRT(R   +R)
   52 SN(I)      = R*SIN(.5*ANGL)
      A          = SLOPT(KK)*(XS(N,KK)   -X1)
      B          = 1./(XS(N,KK)   -X1)
      ANGL       = 0.
      U          = 1.
      V          = 0.
      M          = I2   +1
      DO 54  I=M,NX
      XX         = SS*AO(I)**2
      D          = B*(XX   -X1)
      YY         = YS(N,KK)   +A*ALOG(D)/D
      R          = SQRT(XX**2   +YY**2)
      ANGL       = ANGL   +ATAN2((U*YY   -V*XX),(U*XX   +V*YY))
      U          = XX
      V          = YY
      R          = S*SQRT(R   +R)
   54 SN(I)      = R*SIN(.5*ANGL)
      DO 62  I=2,NX
   62 SO(I,K)    = SO(I,K)   +RR*SN(I)
      IF  (KK.EQ.K2) GO TO 71
      KK         = K2
      RR         = R2
      GO TO 41
   71 SS         = SSO
      IF  (FIX.EQ.0.)
     1SS         = (R1*XS(1,K1)   +R2*XS(1,K2))/(AO(I1)**2   -SO(I1,K)**2)
      SCAL(K)    = SS   +SS
      ITE1(K)    = I1
      ITE2(K)    = I2
      ZV(K)      = ZO(K)   -TYAW*(XO(K)   +SS*AO(I1)*AO(I1))
      DO 72  I=I1,I2
   72 IV(I,K)    = 2
      M          = I1   -1
      DO 74  I=1,M
      ZZ         = ZO(K)   -TYAW*(XO(K)   +SS*AO(I)*AO(I))
      IF  (ZZ.GE.ZV(KTE1)) IV(I,K) = IVO
   74 CONTINUE
      M          = I2   +1
      DO 76  I=M,MX
      ZZ         = ZO(K)   -TYAW*(XO(K)   +SS*AO(I)*AO(I))
      IF  (ZZ.GE.ZV(KTE1)) IV(I,K) = IVO
   76 CONTINUE
      K2         = K2   -1
      K          = K   +1
      IF  (K.LE.KTE2) GO TO 21
      K1         = 2
      K2         = NZ
```

```
      IF (KSYM.EQ.0) GC TC 81
      K1          = 3
      K2          = NZ   +2
   81 SCAL(K)     = SCAL(KTE2)
      DO 82 I=1,MX
      ZZ          = ZO(K)   -TYAW*(XO(K)   +SS*AO(I)*AO(I))
      IF (ZZ.LE.ZS(NS).AND.ZZ.GE.ZV(KTE1)) IV(I,K) = IVO
   82 CONTINLE
      K           = K   +1
      IF (K.LE.K2) GO TO 81
      SCAL(K)     = SCAL(KTE2)
      N           = KTE2
      IF (YAW.LE.0.) GO TC 93
      IO          = ITE1(KTE2)   +1
      DO 92 I=IO,LX
      N           = N   +1
   92 ZV(N)       = ZO(KTE2)   -TYAW*(XO(KTE2)   +SS*AO(I)*AO(I))
   93 I           = ITE1(KTE1)
      ZV(KTE1-1)  =  ZO(KTE1-1)   -TYAW*(XC(KTE1)   +SS*AO(I)*AO(I))
      ZV(N+1)     =  ZO(KTE2+1)
      DO 102 K=K1,K2
      DO 104 I=2,NX
      IF (IV(I,K).GT.0) GC TC 104
      IF (IV(I+1,K+1).GT.C.OR.IV(I-1,K+1).GT.C) IV(I,K) = IV1
      IF (IV(I+1,K-1).GT.0.OR.IV(I-1,K-1).GT.C) IV(I,K) = IV1
  104 CONTINLE
  102 IF (SO(LX,K).LT.1.E-05) IV(LX,K) = 0
      IF (KSYM.NE.0) RETURN
      N           = KTE1   -1
      DO 112 K=1,N
  112 SCAL(K)     = SCAL(KTE1)
      RETURN
      END
```

```
C**SUBROUTINE ESTIM*****************************************************
      SUBROUTINE  ESTIM
C
C     INITIAL ESTIMATE OF REDUCED POTENTIAL
      COMMON       G(161,18,3),SO(161,35),VORT(115),ZV(115),
     1             IV(161,35),ITE1(35),ITE2(35),
     2             AO(161),BO(18),XO(35),YC(35),ZC(35),SCAL(35),
     3             NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,FUS,
     4             YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
      MX           = NX   +1
      MY           = NY   +2
      MZ           = NZ   +3
      DO  12  I=1,161
      DO  12  J=1,18
      DO  12  K=1,3
   12 G(I,J,K)     = O.
      DO  22  K=1,MZ
      WRITE  (N3)  ((G(I,J,1),I=1,MX),J=1,MY)
      WRITE  (N1)  ((G(I,J,1),I=1,MX),J=1,MY)
   22 CONTINUE
      K1           = KTE1   -1
      K2           = KTE2   +ITE2(KTE2)   -NX/2
      DO  32  K=K1,K2
   32 VORT(K)      = 0.
      IO           = 1
      RETURN
      END
```

```
C**SUBROUTINE MIXFLO*****************************************************
      SUBROUTINE  MIXFLO
C     SOLUTION OF EQUATIONS FOR MIXED SUBSONIC AND SUPERSONIC FLOW
C     USING FINITE VOLUME SCHEME
      COMMON         G(161,18,3),SO(161,35),VORT(115),ZV(115),
     1               IV(161,35),ITE1(35),ITE2(35),
     2               AO(161),BO(18),XO(35),YO(35),ZO(35),SCAL(35),
     3               NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,FUS,
     4               YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
      COMMON/SPA/ GL(161,18),QQL(161,18),FL(161,18),
     1               UL(161,18),VL(161,18),WL(161,18),
     2               AL(161,18),BL(161,18),CL(161,18),
     3               RESL(161,18)
      COMMON/FLO/ P1,P2,P3,FRES,IRES,JRES,KRES,ARES,DG,IG,JG,KG,AG,NSUP
      COMMON/SWP/ K,NV,LX,MX,KY,MY,J1,K1,FMACH2,AAO,Q1,Q2,RV,TYAW,TOT
      LX           = NX/2   +1
      MX           = NX    +1
      KY           = NY    +1
      MY           = NY    +2
      J1           = 2
      IF  (FMACH.GE.1.) J1 = 3
      TYAW         = SYAW/CYAW
      FMACH2       = FMACH**2
      AAO          = 1./FMACH**2   +.2
      C1           = 2./P1
      Q2           = 1./P2   -1.
      TOT          = 0.
      FRES         = 0.
      ARES         = 0.
      DG           = 0.
      AG           = 0.
      NSUP         = 0
      K1           = 3
      K2           = NZ    +2
      IF  (KSYM.EQ.1)  GO TO 1
      K1           = 2
      IF  (FMACH.GE.1.) K1 = 3
      K2           = NZ
    1 DO 2 M=2,3
      READ  (N1,ERR=101)  ((G(I,J,M),I=1,MX),J=1,MY)
    2 CONTINUE
      K            = 1
      NV           = KTE1   -1
      RV           = 2.
      DO 12  J=1,MY
      DO 12  I=1,MX
      G(I,J,1)     = G(I,J,2)
      GL(I,J)      = G(I,J,2)
```

```
      QQL(I,J)   = 0.
      FL(I,J)    = 0.
      UL(I,J)    = 0.
      VL(I,J)    = 0.
      WL(I,J)    = 0.
      AL(I,J)    = 0.
      BL(I,J)    = 0.
      CL(I,J)    = 0.
   12 RESL(I,J) = 0.
      IF (KSYM.NE.0) GO TC 21
      CALL YSWEEP
      RV         = 1.
      GO TO 51
   21 DO 22 J=1,MY
      DO 22 I=1,NX
      G(I,J,2)   = G(I,J,3)
   22 GL(I,J)    = G(I,J,3)
      READ (N1,ERR=101) ((G(I,J,3),I=1,MX),J=1,MY)
      WRITE (N2) ((G(I,J,1),I=1,MX),J=1,MY)
      K          = K   +1
      GO TO 51
   31 CALL YSWEEP
      RV         = 1.
      IF (K.NE.KTE2.OR.YAW.LE.0.) GO TO 51
      IO         = ITE1(K)   +1
      DO 42 I=IO,LX
      M          = NX   +2   -I
      V          = G(M,KY,1)   -G(I,KY,1)
      NV         = NV   +1
   42 VORT(NV)   = VORT(NV)   +P3*(V   -VORT(NV))
   51 IF (K.EQ.K2) GO TO 61
      DO 52 J=1,MY
      DO 52 I=1,MX
      G(I,J,1)   = G(I,J,2)
   52 G(I,J,2)   = G(I,J,3)
      READ (N1,ERR=101) ((G(I,J,3),I=1,MX),J=1,MY)
      WRITE (N2) ((G(I,J,1),I=1,MX),J=1,MY)
      K          = K   +1
      GO TO 31
   61 DO 62 M=2,3
      WRITE (N2) ((G(I,J,M),I=1,MX),J=1,MY)
   62 CONTINUE
      FRES       = FRES/64.
      ARES       = ARES/(64.*TOT)
      AG         = AG/TOT
      IO         = 1
      RETURN
  101 IO         = 0
```

```
C**SUBROUTINE YSWEEP*******************************************************
      SUBROUTINE  YSWEEP
C     ROW RELAXATION
C     FINITE VOLUME SCHEME
      COMMON       G(161,18,3),SO(161,35),VORT(115),ZV(115),
     1             IV(161,35),ITE1(35),ITE2(35),
     2             AO(161),BO(18),XO(35),YC(35),ZO(35),SCAL(35),
     3             NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,FUS,
     4             YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IC
      COMMON/SPA/  GL(161,18),QQL(161,18),FL(161,18),
     1             UL(161,18),VL(161,18),WL(161,18),
     2             AL(161,18),BL(161,18),CL(161,18),
     3             RESL(161,18)
      COMMON/FLO/  P1,P2,P3,FRES,IRES,JRES,KRES,ARES,CG,IG,JG,KG,AG,NSUP
      COMMON/SWP/  K,NV,LX,MX,KY,MY,J1,K1,FMACH2,AAO,Q1,Q2,RV,TYAW,TOT
      DIMENSION    X(161),Y(161),Z(161),XM(161),YM(161),ZM(161),
     1             XR(161),YR(161),ZR(161),XRM(161),YRM(161),ZRM(161),
     2             QQM(161),QCP(161),DM(161),DP(161),FM(161),FP(161),
     3             UM(161),UP(161),VM(161),VP(161),WM(161),WP(161),
     4             AM(161),AP(161),BM(161),BP(161),CM(161),CP(161),
     5             ABM(161),AEP(161),BCM(161),ECP(161)
      DIMENSICN    CAM(161),CAP(161),ABCM(161),ABCF(161),
     1             RESO(161),QA(161),P(161),Q(161),R(161),QP(161),
     2             FU(161),FV(161),FW(161),FUU(161),FVV(161),FWW(161),
     3             RES(161),D(161),E(161),CG(161)
      PI         = 3.1415926535898
      BV         = .125*RV
      CV         = .125*(2.   -RV)
      D(1)       = 0.
      E(1)       = 0.
      I2         = NX
      J          = 1
      XS         = XO(K)/SCAL(KTE1)
      YS         = YO(K)/SCAL(KTE1)
      ZS         = ZO(K)/SCAL(KTE1)
      XRS        = XO(K+1)/SCAL(KTE1)
      YRS        = YO(K+1)/SCAL(KTE1)
      ZRS        = ZO(K+1)/SCAL(KTE1)
      S2         = SCAL(K)/SCAL(KTE1)
      S1         = .5*S2
      SR2        = SCAL(K+1)/SCAL(KTE1)
      SR1        = .5*SR2
      FS         = FUS/SCAL(KTE1)
      DO 12  I=1,MX
      RESO(I)    = 1.
      QA(I)      = 0.
      P(I)       = 0.
      Q(I)       = 0.
```

```
      R(I)       = 0.
      QP(I)      = 0.
      FU(I)      = 0.
      FV(I)      = 0.
      FW(I)      = 0.
      CG(I)      = 0.
      XM(I)      = XS   +S1*(AO(I)**2  -(BO(J)   +SO(I,K))**2)
      YM(I)      = YS   +S2*AO(I)*(BO(J)   +SO(I,K))
      ZM(I)      = ZS
      XRM(I)     = XRS  +SR1*(AO(I)**2  -(BO(J)   +SO(I,K+1))**2)
      YRM(I)     = YRS  +SR2*AO(I)*(BO(J)   +SO(I,K+1))
   12 ZRM(I)     = ZRS
      IF (FUS.LE.C.) GO TO 21
      DO 14 I=1,MX
      A          = .25*(ZM(I)**2  -YM(I)**2)   +FS**2
      B          = .5*ZM(I)*YM(I)
      S          = SQRT(A**2  +B**2)
      T          = 0.
      IF (S.GT.0.) T = .5*ATAN2(B,A)
      IF (B.EC.0..AND.YM(I).GT.(FS   +FS)) T = .5*PI
      IF (B.EC.C..AND.YM(I).LT.-(FS   +FS)) T = -.5*PI
      S          = SQRT(S)
      YM(I)      = .5*YM(I)   +S*SIN(T)
      ZM(I)      = .5*ZM(I)   +S*COS(T)
      A          = .25*(ZRM(I)**2  -YRM(I)**2)   +FS**2
      B          = .5*ZRM(I)*YRM(I)
      T          = 0.
      S          = SQRT(A**2  +B**2)
      IF (S.GT.0.) T = .5*ATAN2(B,A)
      S          = SQRT(S)
      YRM(I)     = .5*YRM(I)   +S*SIN(T)
   14 ZRM(I)     = .5*ZRM(I)   +S*COS(T)
   21 DO 22 I=1,MX
      X(I)       = XM(I)
      Y(I)       = YM(I)
      Z(I)       = ZM(I)
      XR(I)      = XRM(I)
      YR(I)      = YRM(I)
      ZR(I)      = ZRM(I)
      XM(I)      = XS   +S1*(AO(I)**2  -(BC(J+1)   +SO(I,K))**2)
      YM(I)      = YS   +S2*AO(I)*(BO(J+1)   +SC(I,K))
      ZM(I)      = ZS
      XRM(I)     = XRS  +SR1*(AO(I)**2  -(BO(J+1)   +SO(I,K+1))**2)
      YRM(I)     = YRS  +SR2*AO(I)*(BO(J+1)   +SO(I,K+1))
   22 ZRM(I)     = ZRS
      IF (FUS.LE.C.) GO TO 31
      DO 24 I=1,MX
      A          = .25*(ZM(I)**2  -YM(I)**2)   +FS**2
```

```
      B           = .5*ZM(I)*YM(I)
      S           = SQRT(A**2   +B**2)
      T           = 0.
      IF (S.GT.0.) T  = .5*ATAN2(B,A)
      IF (B.EQ.0..AND.YM(I).GT.(FS   +FS)) T = .5*PI
      IF (B.EQ.0..AND.YM(I).LT.-(FS   +FS)) T = -.5*PI
      S           = SQRT(S)
      YM(I)       = .5*YM(I)   +S*SIN(T)
      ZM(I)       = .5*ZM(I)   +S*COS(T)
      A           = .25*(ZRM(I)**2   -YRM(I)**2)   +FS**2
      B           = .5*ZRM(I)*YRM(I)
      T           = 0.
      S           = SQRT(A**2   +B**2)
      IF (S.GT.0.) T  = .5*ATAN2(B,A)
      S           = SQRT(S)
      YRM(I)      = .5*YRM(I)   +S*SIN(T)
   24 ZRM(I)      = .5*ZRM(I)   +S*COS(T)
   31 DO 32  I=1,NX
      XX          = XR(I+1)   -XR(I)   +XRM(I+1)   -XRM(I)
     1              +X(I+1)   -X(I)   +XM(I+1)   -XM(I)
      XY          = XR(I+1)   +XR(I)   -XRM(I+1)   -XRM(I)
     1              +X(I+1)   +X(I)   -XM(I+1)   -XM(I)
      XZ          = XR(I+1)   +XR(I)   +XRM(I+1)   +XRM(I)
     1              -(X(I+1)   +X(I)   +XM(I+1)   +XM(I))
      YX          = YR(I+1)   -YR(I)   +YRM(I+1)   -YRM(I)
     1              +Y(I+1)   -Y(I)   +YM(I+1)   -YM(I)
      YY          = YR(I+1)   +YR(I)   -YRM(I+1)   -YRM(I)
     1              +Y(I+1)   +Y(I)   -YM(I+1)   -YM(I)
      YZ          = YR(I+1)   +YR(I)   +YRM(I+1)   +YRM(I)
     1              -(Y(I+1)   +Y(I)   +YM(I+1)   +YM(I))
      ZX          = ZR(I+1)   -ZR(I)   +ZRM(I+1)   -ZRM(I)
     1              +Z(I+1)   -Z(I)   +ZM(I+1)   -ZM(I)
      ZY          = ZR(I+1)   +ZR(I)   -ZRM(I+1)   -ZRM(I)
     1              +Z(I+1)   +Z(I)   -ZM(I+1)   -ZM(I)
      ZZ          = ZR(I+1)   +ZR(I)   +ZRM(I+1)   +ZRM(I)
     1              -(Z(I+1)   +Z(I)   +ZM(I+1)   +ZM(I))
      FXX         = YY*ZZ   -YZ*ZY
      FYX         = YZ*ZX   -YX*ZZ
      FZX         = YX*ZY   -YY*ZX
      FXY         = ZY*XZ   -ZZ*XY
      FYY         = ZZ*XX   -ZX*XZ
      FZY         = ZX*XY   -ZY*XX
      FXZ         = XY*YZ   -XZ*YY
      FYZ         = XZ*YX   -XX*YZ
      FZZ         = XX*YY   -XY*YX
      FM(I)       = FXX*XX   +FYX*XY   +FZX*XZ
      A           = 1./FM(I)
      GX          = G(I+1,J,2)   -G(I,J,2)   +G(I+1,J+1,2)   -G(I,J+1,2)
```

```
      GXR       = G( I+1, J,3)  -G(I,J,3)  +G(I+1,J+1,3)  -G(I ,J+1,3)
      GY        = G( I+1, J,2)  +G(I,J,2)  -G(I+1,J+1,2)  -G(I ,J+1,2)
      GYR       = G( I+1, J,3)  +G(I,J,3)  -G(I+1,J+1,3)  -G(I ,J+1,3)
      GXY       = G( I+1, J,2)  -G(I,J,2)  -G(I+1,J+1,2)  +G(I ,J+1,2)
      GXYR      = G( I+1, J,3)  -G(I,J,3)  -G(I+1,J+1,3)  +G(I ,J+1,3)
      GZ        = G( I+1, J,3)  +G(I,J,3)  +G(I+1,J+1,3)  +G(I ,J+1,3)
     1           -(G(I+1,J,2)  +G(I ,J,2)  +G(I+1,J+1,2)  +G(I,J+1,2))
      ABM(I)    = GXYR  +GXY
      BCM(I)    = GYR  -GY
      CAM(I)    = GXR  -GX
      ABCM(I)   = GXYR  -GXY
      GX        = GXR  +GX
      GY        = GYR  +GY
      U         = (FXX*GX  +FYX*GY  +FZX*GZ)*A  +CA
      V         = (FXY*GX  +FYY*GY  +FZY*GZ)*A  +SA
      W         = (FXZ*GX  +FYZ*GY  +FZZ*GZ)*A  +SYAW
      QQM(I)    = U*U  +V*V  +W*W
      AA        = AA0  -.2*QQM(I)
      DM(I)     = ABS(FMACH2*AA)
      UM(I)     = FXX*U  +FXY*V  +FXZ*W
      VM(I)     = FYX*U  +FYY*V  +FYZ*W
      WM(I)     = FZX*U  +FZY*V  +FZZ*W
      AM(I)     = (FXX**2  +FXY**2  +FXZ**2  -UM(I)**2/AA)*A
      BM(I)     = (FYX**2  +FYY**2  +FYZ**2  -VM(I)**2/AA)*A
   32 CM(I)     = (FZX**2  +FZY**2  +FZZ**2  -WM(I)**2/AA)*A
      DO 34 I=1,NX
   34 DM(I)     = DM(I)*DM(I)*SQRT(DM(I))
      DO 36 I=1,NX
      UM(I)     = DM(I)*UM(I)
      VM(I)     = DM(I)*VM(I)
      WM(I)     = DM(I)*WM(I)
      AM(I)     = DM(I)*AM(I)
      BM(I)     = DM(I)*BM(I)
      CM(I)     = DM(I)*CM(I)
      FM(I)     = DM(I)*FM(I)
      ABM(I)    = (AM(I)  +BM(I))*ABM(I)
      BCM(I)    = (BM(I)  +CM(I))*BCM(I)
      CAM(I)    = (CM(I)  +AM(I))*CAM(I)
   36 ABCM(I)   = (AM(I)  +BM(I)  +CM(I))*ABCM(I)
      IF (J.LT.2) GO TO 71
   41 DO 42 I=2,NX
      QQR       = QQP(I)  +QQP(I-1)  +QQM(I)  +QQM(I-1)
      FR        = FP(I)  +FP(I-1)  +FM(I)  +FM(I-1)
      UR        = UP(I)  +UP(I-1)  +UM(I)  +UM(I-1)
      VR        = VP(I)  +VP(I-1)  +VM(I)  +VM(I-1)
      WR        = WP(I)  +WP(I-1)  +WM(I)  +WM(I-1)
      QQ        = RESQ(I)*BV*(QQR  +QQL(I,J))
      F         = BV*(FR  +FL(I,J))
```

```
      U            = BV*(UR   +UL(I,J))
      V            = BV*(VR   +VL(I,J))
      W            = CV*(WR   +WL(I,J))
      QQL(I,J)     = QQR
      FL(I,J)      = FR
      UL(I,J)      = UR
      VL(I,J)      = VR
      WL(I,J)      = WR
      AA           = AAO  -.2*CC
      QA(I)        = QQ/AA
      A            = 16.*RESO(I)/(F*AA)
      QP(I)        = Q(I)
      P(I)         = 0.
      Q(I)         = 0.
      R(I)         = 0.
      F            = A*AMAX1(ABS(U),ABS(V),ABS(W))
      FU(I)        = F*U
      FV(I)        = F*ABS(V)
      FW(I)        = F*ABS(W)
      IF  (QA(I).LE.1.) GO TO 42
      NSUP         = NSUP  +1
      F            = A*(1.   -1./QA(I))
      FUU(I)       = F*U*U
      FVV(I)       = F*V*V
      FWW(I)       = F*W*W
      GXX          = G(I+1,J,2)  -G(I,J,2)   -G(I,J,2)   +G(I-1,J,2)
      GYY          = G(I,J-1,2)  -G(I,J,2)   -G(I,J,2)   +G(I,J+1,2)
      GZZ          = G(I,J,3)  -G(I,J,2)   -G(I,J,2)   +GL(I,J)
      GXY          = G(I+1,J-1,2)  -G(I-1,J-1,2)
     1               -G(I+1,J+1,2)  +G(I-1,J+1,2)
      GYZ          = G(I,J-1,3)  -G(I,J+1,3)  -GL(I,J-1)  +GL(I,J+1)
      GZX          = G(I+1,J,3)  -G(I-1,J,3)  -GL(I+1,J)  +GL(I-1,J)
      F            = .25*F
      FXY          = F*U*V*GXY
      FYZ          = F*V*W*GYZ
      FZX          = F*W*U*GZX
      P(I)         = FUU(I)*GXX  +FXY   +FZX
      Q(I)         = FVV(I)*GYY  +FYZ   +FXY
      R(I)         = FWW(I)*GZZ  +FZX   +FYZ
   42 CONTINUE
      PF           = -P(2)
      DO 52  I=2,NX
      AV           = RESC(I)*RV
      PB           = PF
      PF           = P(I)
      IF  ((FU(I)  +FU(I+1)).LT.0.) PF = -P(I+1)
      A            = UP(I)  -UP(I-1)  +UM(I)  -UM(I-1)
     1               +VP(I)  +VP(I-1)  -VM(I)  -VM(I-1)
```

```
     2          -ABP(I)   +ABP(I-1)   +ABM(I)   -ABM(I-1)
      B       = WP(I)   +WP(I-1)   +WM(I)   +WM(I-1)
     1          -BCP(I)   -BCP(I-1)   +BCM(I)   +BCM(I-1)
     2          -CAP(I)   +CAP(I-1)   -CAM(I)   +CAM(I-1)
     3          +ABCP(I)   -ABCP(I-1)   -ABCM(I)   +ABCM(I-1)   +R(I)
      RES(I)  = AV*(A   +B   +PF   -PB   +GP(I)   -Q(I))   +RESL(I,J)
      RESL(I,J) = A   -B
      AR      = AP(I)   +AP(I-1)   +AM(I)   +AM(I-1)
      BR      = BP(I)   +BP(I-1)   +BM(I)   +BM(I-1)
      CR      = CP(I)   +CP(I-1)   +CM(I)   +CM(I-1)
      A       = AV*AR   +AL(I,J)
      B       = AV*BR   +BL(I,J)
      C       = AV*CR   +CL(I,J)
      AL(I,J) = AR
      BL(I,J) = BR
      CL(I,J) = CR
      A       = A   +A
      B       = B   +B
      C       = C   +C
      TP      = A
      T       = A   +A   +Q1*(B   +C)   +Q2*(ABS(FU(I))   +FV(I)   +FW(I))
      S       = (B   +Q2*FV(I))*CG(I)
     1          +(C   +Q2*FW(I))*(G(I,J,1)   -GL(I,J))
      F       = 0.
      IF (QA(I).LE.1.) GO TO 53
      F       = FUU(I)
      TP      = TP   +F
      T       = T   +3.*(F   +FVV(I)   +FWW(I))   +F
      S       = S   +3.*(FVV(I)*CG(I)   +FWW(I)*(G(I,J,1)   -GL(I,J)))
   53 TM      = TP
      IF (FU(I).GE.0.) TM = TM   +Q2*FU(I)   +F   +F
      IF (FU(I).LT.0.) TP = TP   -Q2*FU(I)   +F   +F
      TM      = TM*RESO(I-1)
      B       = 1./(T   -TM*E(I-1))
      D(I)    = B*(RES(I)   +S   +TM*D(I-1))
   52 E(I)    = B*TP
      IF (J.LT.J1.OR.K.LT.K1) GO TO 71
      I       = I2
      CG(I+1) = 0.
      DO 62 M=2,I2
      GL(I,J) = G(I,J,2)
      CG(I)   = D(I)   +E(I)*CG(I+1)
      G(I,J,2) = G(I,J,2)   +CG(I)
      TOT     = TOT   +1.
      ARES    = ARES   +ABS(RES(I))
      IF (ABS(RES(I)).LE.ABS(FRES)) GO TO 63
      FRES    = RES(I)
      IRES    = I
```

```
      JRES        = J
      KRES        = K
   63 AG          = AG   +ABS(CG(I))
      IF  (ABS(CG(I)).LE.ABS(DG)) GO TO 62
      DG          = CG(I)
      IG          = I
      JG          = J
      KG          = K
   62 I           = I   -1
   71 DO  72  I=1,NX
      QQP(I)      = QQM(I)
      DP (I)      = DM (I)
      FP (I)      = FM (I)
      UP (I)      = UM (I)
      VP (I)      = VM (I)
      WP (I)      = WM (I)
      AP (I)      = AM (I)
      BP (I)      = BM (I)
      CP (I)      = CM (I)
      ABP(I)      = ABM(I)
      BCP(I)      = BCM(I)
      CAP(I)      = CAM(I)
   72 ABCP(I)     = ABCM(I)
      J           = J   +1
      IF  (J   -KY) 21,81,101
   81 I2          = ITE2(K)
      IF  (ITE2(K).EQ.MX)  I2 = LX
      IF  (ISYM.EQ.1)  I2 = NX
      DO  82  I=1,NX
      IF  (ISYM.EQ.1)  GO TO 83
      IF  (IV(I,K).EQ.2.OR.IV(I+1,K).EQ.2) GO TO 83
      M           = NX   +1   -I
      QQM(I)      = QQP(M)
      DM (I)      = DP (M)
      FM (I)      = FP (M)
      UM (I)      = -UP(M)
      VM (I)      = -VP(M)
      WM (I)      = WP (M)
      AM (I)      = AP (M)
      BM (I)      = BP (M)
      CM (I)      = CP (M)
      ABM(I)      = ABP(M)
      BCM(I)      = -BCP(M)
      CAM(I)      = -CAF(M)
      ABCM(I)     = ABCP(M)
      GO TO 82
   83 QQM(I)      = QQP(I)
      DM (I)      = DP (I)
```

```
      FM(I)      = FP(I)
      UM(I)      = UP(I)
      VM(I)      = -VP(I)
      WM(I)      = WP(I)
      AM(I)      = AP(I)
      BM(I)      = BP(I)
      CM(I)      = CP(I)
      ABM(I)     = -ABP(I)
      BCM(I)     = -BCF(I)
      CAM(I)     = CAP(I)
      ABCM(I)    = -ABCP(I)
   82 CONTINUE
      DO 92 I=2,NX
      IF (IABS(IV(I,K)).GT.1) GO TO 92
      RESO(I)    = 0.
      A          = 1.   -AMAXO(0,IABS(IV(I,K)))
      S          = A*(G(I+1,J,2)  -G(I,J,2)  -G(I,J,2)  +G(I-1,J,2))
     1             +G(I,J-1,2)  -G(I,J,2)  -G(I,J,2)  +G(I,J+1,2)
      RESL(I,J)  = S   +S
      AL(I,J)    = A
      BL(I,J)    = 1.
      CL(I,J)    = 0.
   92 CONTINUE
      GO TO 41
  101 S1         = .5*SCAL(K)
      I          = NX   +2   -I2
      I1         = ITE1(K)
      N          = NV
      IF (I.NE.I1.OR.ISYM.EQ.1) GO TO 103
      V          = G(I2,KY,2)  -G(I1,KY,2)
      NV         = NV   +1
      VORT(NV)   = VORT(NV)   +P3*(V   -VORT(NV))
      N          = NV
  103 I          = I   -1
      V          = 0.
      IF (IV(I,K).NE.1) GO TO 109
      ZZ         = ZO(K)   -TYAW*(XO(K)   +S1*AO(I)*AO(I))
  105 IF (ZZ.GE.ZV(N-1)) GO TO 107
      N          = N   -1
      GO TO 105
  107 A          = (ZZ   -ZV(N-1))/(ZV(N)   -ZV(N-1))
      V          = A*VORT(N)   +(1.   -A)*VORT(N-1)
  109 M          = NX   +2   -I
      G(I,KY+1,2) = G(M,KY-1,2)   -V
      G(M,KY+1,2) = G(I,KY-1,2)   +V
      G(M,KY,2)   = G(I,KY,2)   +V
      IF (I.GT.1) GO TO 103
      G(I,KY,2)   = -.5*V
```

```
G(M,KY,2)     =  .5*V
G(LX,KY+1,2)  =  G(LX,KY-1,2)
RETURN
END
```

```
C**SUBROUTINE VELC*************************************************************
      SUBROUTINE  VELC  (K,SV,SM,CP,XP,YP,XMAX,XMIN,YMAX,YMIN)
C     CALCULATES SURFACE VELOCITY
      COMMON        G(161,18,3),SO(161,35),VORT(115),ZV(115),
     1              IV(161,35),ITE1(35),ITE2(35),
     2              AO(161),BO(18),XO(35),YO(35),ZO(35),SCAL(35),
     3              NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,FUS,
     4              YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
      COMMON/SPA/ XL(161,18),YL(161,18),ZL(161,18),
     1              X(161,18),Y(161,18),Z(161,18),
     2              XR(161,18),YR(161,18),ZR(161,18),
     3              RESL(161,18)
      COMMON /UVW/ UU(161),VV(161),WW(161)
      DIMENSION     SV(1),SM(1),CP(1),XP(1),YP(1)
      PI            = 3.14159265358 9E
      Q1            = .2*FMACH**2
      T1            = 1./(.7*FMACH**2)
      MX            = NX   +1
      NMAX          = 1
      AV            = .5
      IF (KSYM.EQ.1.AND.K.EQ.KTE1) GO TO 1
      NMAX          = 2
      AV            = .25
    1 N             = K
      IF (K.NE.KTE1) GO TO 11
      N             = KTE1  -2  +KSYM
      DO 2 J=1,2
      DO 2 I=1,MX
      X(I,J)        = 0.
      Y(I,J)        = 0.
      Z(I,J)        = 0.
      XR(I,J)       = 0.
      YR(I,J)       = 0.
    2 ZR(I,J)       = 0.
   11 N             = N   +1
      XRS           = XO(N)/SCAL(KTE1)
      YRS           = YO(N)/SCAL(KTE1)
      ZRS           = ZO(N)/SCAL(KTE1)
      S2            = SCAL(N)/SCAL(KTE1)
      S1            = .5*S2
      FS            = FUS/SCAL(KTE1)
      DO 12 J=1,2
      M             = NY   +2   -J
      DO 12 I=1,MX
      XL(I,J)       = X(I,J)
      YL(I,J)       = Y(I,J)
      ZL(I,J)       = Z(I,J)
      X(I,J)        = XR(I,J)
```

```
      Y(I,J)      = YR(I,J)
      Z(I,J)      = ZR(I,J)
      XR(I,J)     = XRS  +S1*(AO(I)**2  -(BO(M)  +SC(I,N))**2)
      YR(I,J)     = YRS  +S2*AO(I)*(BO(M)  +SO(I,N))
   12 ZR(I,J)     = ZRS
      IF (FUS.LE.0.) GO TC 21
      DO 14 J=1,2
      DO 14 I=1,MX
      A           = .25*(ZR(I,J)**2  -YR(I,J)**2)  +FS**2
      B           = .5*ZR(I,J)*YR(I,J)
      S           = SQRT(A**2  +B**2)
      T           = 0.
      IF (S.GT.0.) T = .5*ATAN2(B,A)
      IF (B.EC.0..AND.YR(I,J).GT.(FS  +FS)) T = .5*PI
      IF (B.EC.C..AND.YR(I,J).LT.-(FS  +FS)) T = -.5*PI
      S           = SQRT(S)
      YR(I,J)     = .5*YR(I,J)  +S*SIN(T)
   14 ZR(I,J)     = .5*ZR(I,J)  +S*COS(T)
   21 IF (N.LE.K) GO TC 11
      J           = NY  +1
      DO 22 I=2,NX
      U           = 0.
      V           = 0.
      W           = 0.
      M           = 1
      N           = 1
      XX          = X(I+1,1)  -X(I,1)
      XY          = X(I,2)  -X(I,1)
      XZ          = XR(I,1)  -X(I,1)
      YX          = Y(I+1,1)  -Y(I,1)
      YY          = Y(I,2)  -Y(I,1)
      YZ          = YR(I,1)  -Y(I,1)
      ZX          = Z(I+1,1)  -Z(I,1)
      ZY          = Z(I,2)  -Z(I,1)
      ZZ          = ZR(I,1)  -Z(I,1)
      GX          = G(I+1,J,2)  -G(I,J,2)
      GY          = G(I,J-1,2)  -G(I,J,2)
      GZ          = G(I,J,3)  -G(I,J,2)
   23 FXX         = YY*ZZ  -YZ*ZY
      FYX         = YZ*ZX  -YX*ZZ
      FZX         = YX*ZY  -YY*ZX
      FXY         = ZY*XZ  -ZZ*XY
      FYY         = ZZ*XX  -ZX*XZ
      FZY         = ZX*XY  -ZY*XX
      FXZ         = XY*YZ  -XZ*YY
      FYZ         = XZ*YX  -XX*YZ
      FZZ         = XX*YY  -XY*YX
      F           = 1./(FXX*XX  +FYX*XY  +FZX*XZ)
```

```
   U          = U   +(FXX*GX   +FYX*GY   +FZX*GZ)*F   +CA
   V          = V   +(FXY*GX   +FYY*GY   +FZY*GZ)*F   +SA
   W          = W   +(FXZ*GX   +FYZ*GY   +FZZ*GZ)*F   +SYAW
   IF (M.EQ.2) GO TO 25
   M          = 2
   XX         = X(I,1)   -X(I-1,1)
   YX         = Y(I,1)   -Y(I-1,1)
   ZX         = Z(I,1)   -Z(I-1,1)
   GX         = G(I,J,2)   -G(I-1,J,2)
   GO TO 23
25 IF (N.EQ.NMAX) GO TO 27
   M          = 1
   N          = 2
   XX         = X(I+1,1)   -X(I,1)
   XZ         = X(I,1)   -XL(I,1)
   YX         = Y(I+1,1)   -Y(I,1)
   YZ         = Y(I,1)   -YL(I,1)
   ZX         = Z(I+1,1)   -Z(I,1)
   ZZ         = Z(I,1)   -ZL(I,1)
   GX         = G(I+1,J,2)   -G(I,J,2)
   GZ         = G(I,J,2)   -G(I,J,1)
   GO TO 23
27 U          = AV*U
   V          = AV*V
   W          = AV*W
   UU(I) = U
   VV(I) = V
   WW(I) = W
   Q          = U*U   +V*V   +W*W
   SV(I)      = SQRT(Q)
   Q          = AMAX1(.01,(1.   +Q1*(1.   -Q)))
   SM(I)      = FMACH*SV(I)/SQRT(Q)
   CP(I)      = T1*(Q**3.5   -1.)
   XP(I)      = SCAL(KTE1)*X(I,1)
22 YP(I)      = SCAL(KTE1)*Y(I,1)
   I1         = ITE1(K)
   I2         = ITE2(K)
   XMAX       = SCAL(KTE1)*X(I1,1)
   XMIN       = XMAX
   YMAX       = SCAL(KTE1)*Y(I1,1)
   YMIN       = YMAX
   DO 32 I=I1,I2
   XMAX       = AMAX1(XMAX,XP(I))
   XMIN       = AMIN1(XMIN,XP(I))
   YMAX       = AMAX1(YMAX,YP(I))
32 YMIN       = AMIN1(YMIN,YP(I))
   RETURN
   END
```

```
C**SUBROUTINE CPLCT*****************************************************
      SUBROUTINE  CPLCT (I3,I4,FMACH,XP,YP,CP,SM,I1,I2,KPLOT)
C     PLCTS CF AT COMPUTATIONAL MESH PCINTS
      COMMON /PCKR/  PTCK
      COMMON /PARMT3/ XT3(161), YT3(161), ZT3(161),
     1                UT3(161),VT3(161),WT3(161),NOI,Z
      COMMON /UVW/ UU(161),VV(161),WW(161)
      COMMON /PRS/ XOCD(161)
      COMMON /SFARE/ LINE(90),DUMY(10)
      DIMENSICN KCDE(2),XP(1),YP(1),CP(1),SM(1)
      DATA        KODE/1H ,1H+/
      DATA IST/1H*/
      NOI = C
      IMIN = I1 + (I2 - I1)/2
      CHD = XF(I1) - XP(IMIN)
    2 FORMAT(40H0PLOT OF CP AT CCMPUTATICNAL MESH POINTS/
     1      1CH0       X    ,1CH       Y    ,      7HMACH NO,    8H     CP   ,
     2 7H    XCC ,2X,5HCP* =,F8.4,2X, 7HCHCRD =,F1C.4)
      CPC         = ((1.   +.2*FMACH**2)**3.5  -1.)/(.7*FMACH**2)
      DO 12  I=1,90
   12 LINE(I)     = KODE(1)
      CPS =  (((5. + FMACH**2)/6.)**3.5- 1.)/(.7*FMACH**2)
      IF (KPLCT.EC.0.CR.KPLOT.GT.1) GO TO 15
      WRITE (6,2) CPS,CHD
   15 CONTINUE
      KS = 3C.*(CPO - CPS) + 4.5
      IF (KS.GE.1 .AND. KS.LE.90) LINE(KS)=IST
      DO 22  I=I3,I4
      K           = 30.*(CPO  -CP(I))   +4.5
      K           = MINC(9C,K)
      LINE(K)     = KODE(2)
      XOC = (XF(I)- XP(IMIN))/CHD
      XOCD(I) = XCC
      IF (KPLCT.EC.0.CR.KPLOT.GT.1) GC TC 20
      WRITE (6,61C) XP(I),YP(I),SM(I),CP(I),XCC,LINE
   20 CONTINUE
      LINE(K) = KCDE(1)
      IF (I.LT.I1.OR.I.GT.I2) GO TO 22
      NOI = NCI + 1
      XT3(NOI) = XP(I)
      YT3(NOI) = YP(I)
      ZT3(NOI) = Z
      UT3(NOI) = UU(I)
      VT3(NOI) = VV(I)
      WT3(NOI) = WW(I)
   22 IF (K.EC.KS) LINE(KS) = IST
      IF (KPLCT.EC.0.CR.KPLOT.GT.1) GO TC 25
      CALL INVRT6
```

```
   25 CONTINLE
      RETLRN
  61C FORMAT(2F10.4,F7.4,F8.4,F7.4,90A1)
      ENC
```

```
C**SUBROUTINE INVRT6*****************************************************
      SUBROUTINE INVRT6
      COMMON /PCKR/ PTCK
      COMMON /PARMT3/ XT3(161), YT3(161), ZT3(161),
     1                UT3(161),VT3(161),WT3(161),NOI,Z
      DIMENSION P(161,6),TEMP(161)
      EQUIVALENCE (XT3(1),P(1,1))
      DO 30 J=1,6
      DO 10 I=1,NOI
      M=NOI-I+1
      TEMP(M)=P(I,J)
   10 CONTINUE
      DO 20 I=1,NOI
      P(I,J)=TEMP(I)
   20 CONTINUE
   30 CONTINUE
      WRITE (6,100)  (I,XT3(I),ZT3(I),YT3(I),UT3(I),
     1 WT3(I),VT3(I),I=1,NOI)
      RETURN
  100 FORMAT(4H0 I ,1X,6HXT3(I),4X,6HZT3(I),4X,6HYT3(I),4X,6HUT3(I),
     1                4X,6HWT3(I),4X,6HVT3(I),
     2    6X,4H  I ,1X,6HXT3(I),4X,6HZT3(I),4X,6HYT3(I),4X,6HUT3(I),
     3                4X,6HWT3(I),4X,6HVT3(I)//
     4 (I4,1X,6G10.3,2X,I4,1X,6G10.3))
      END
```

```
C**SUBROUTINE FORCF*******************************************************
      SUBROUTINE  FORCF (I1,I2,XP,YP,CP,AL,CHORD,XM,YM,CL,CD,CM)
C     CALCULATES SECTION FORCE COEFFICIENTS
      DIMENSION   XP(1),YP(1),CP(1)
      RAD         = 57.295779513082
      ALPHA       = AL/RAD
      CL          = 0.
      CD          = 0.
      CM          = 0.
      N           = I2   -1
      DO 12  I=I1,N
      DX          = (XP(I+1)   -XP(I))/CHORD
      DY          = (YP(I+1)   -YP(I))/CHORD
      XA          = (.5*(XP(I+1)   +XP(I))   -XM)/CHORD
      YA          = (.5*(YP(I+1)   +YP(I))   -YM)/CHORD
      CPA         = .5*(CP(I+1)   +CP(I))
      DCL         = -CPA*DX
      DCD         = CPA*DY
      CL          = CL   +DCL
      CD          = CD   +DCD
   12 CM          = CM   +DCD*YA   -DCL*XA
      DCL         = CL*COS(ALPHA)   -CD*SIN(ALPHA)
      CD          = CL*SIN(ALPHA)   +CD*COS(ALPHA)
      CL          = DCL
      RETURN
      END
```

```
C**SUBROUTINE TOTFOR*************************************************************
      SUBROUTINE  TOTFOR(KTE1,KTE2,CHORD,SCL,SCD,SCM,XO,YPO,ZPO,
     1                   CL,CD,CMP,CMR,CMY)
C     CALCULATES TOTAL FORCE COEFFICIENTS
      DIMENSION   CHORD(1),SCL(1),SCD(1),SCM(1),XO(1),YPO(1),ZPO(1)
      SPAN        = ZPO(KTE2)   -ZPO(KTE1)
      CL          = 0.
      CD          = 0.
      CMP         = 0.
      CMR         = 0.
      CMY         = 0.
      S           = 0.
      K           = KTE1
      QL          = SCL(K)*CHORD(K)
      QD          = SCD(K)*CHORD(K)
      QM          = CHORD(K)*(SCM(K)*CHORD(K)
     1                  -SCL(K)*XO(K)   +SCD(K)*YPO(K))
   11 DZ          = .5*(ZPO(K+1)   -ZPO(K))
      AZ          = .5*(ZPO(K+1)   +ZPO(K))
      PL          = SCL(K+1)*CHORD(K+1)
      PD          = SCD(K+1)*CHORD(K+1)
      PM          = CHORD(K+1)*(SCM(K+1)*CHORD(K+1)
     1                  -SCL(K+1)*XO(K+1)   +SCD(K+1)*YPO(K+1))
      CLA         = DZ*(PL   +QL)
      CDA         = DZ*(PD   +QD)
      CL          = CL   +CLA
      CD          = CD   +CDA
      CMP         = CMP   +DZ*(PM   +QM)
      CMR         = CMR   +AZ*CLA
      CMY         = CMY   +AZ*CDA
      S           = S   +DZ*(CHORD(K+1)   +CHORD(K))
      QL          = PL
      QD          = PD
      QM          = PM
      K           = K   +1
      IF  (K.LT.KTE2)  GO TO 11
      CL          = CL/S
      CD          = CD/S
      CMP         = CMP*SPAN/S**2
      CMR         = (CMR   +CMR)/(S*SPAN)
      CMY         = (CMY   +CMY)/(S*SPAN)
      RETURN
      END
```

```
C**SUBROUTINE REFIN*****************************************************
      SUBROUTINE  REFIN
C     HALVES MESH SIZE
      COMMON      G(161,18,3),SO(161,35),VORT(115),ZV(115),
     1            IV(161,35),ITE1(35),ITE2(35),
     2            AO(161),BO(18),XO(35),YO(35),ZO(35),SCAL(35),
     3            NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,FUS,
     4            YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
      MX          = NX   +1
      KY          = NY   +1
      MY          = NY   +2
      MZ          = NZ   +3
      MXO         = NX/2   +1
      MYO         = NY/2   +2
      MZO         = NZ/2   +1
      K           = 1
      IF (KSYM.EQ.0) GO TO 11
      MZO         = NZ/2   +3
      READ (N1,ERR=401) ((G(I,J,1),I=1,MXO),J=1,MYO)
      K           = 2
   11 READ (N1,ERR=401) ((G(I,J,1),I=1,MXO),J=1,MYO)
      J           = NY/2   +1
      JJ          = KY
   21 I           = MXO
      II          = MX
   31 G(II,JJ,1)  =  G(I,J,1)
      I           = I   -1
      II          = II   -2
      IF  (I.GT.0) GO  TO 31
      J           = J   -1
      JJ          = JJ   -2
      IF  (J.GT.0) GO  TO 21
      DO  42  J=1,KY,2
      DO  42  I=2,NX,2
   42 G(I,J,1)   = .5*(G(I+1,J,1)  +G(I-1,J,1))
      DO  52  I=1,MX
      DO  54  J=2,NY,2
   54 G(I,J,1)   = .5*(G(I,J+1,1)  +G(I,J-1,1))
   52 G(I,MY,1) = 0.
      G(1,NY,1)    =  0.
      G(MX,NY,1)   =  0.
      WRITE  (N2) ((G(I,J,1),I=1,MX),J=1,MY)
      K           = K   +1
      IF  (K.LE.MZO) GO TO 11
      REWIND N1
      REWIND N2
      READ (N2,ERR=401) ((G(I,J,1),I=1,MX),J=1,MY)
      READ (N2,ERR=401) ((G(I,J,3),I=1,MX),J=1,MY)
```

```
      WRITE (N1) ((G(I,J,1),I=1,MX),J=1,MY)
      K          = 1
      IF (KSYM.NE.0) K = 2
  111 K          = K   +1
      DO 112 J=1,MY
      DO 112 I=1,MX
  112 G(I,J,2)   = .5*(G(I,J,1)   +G(I,J,3))
      DO 122 M=2,3
      WRITE (N1) ((G(I,J,M),I=1,MX),J=1,MY)
  122 CONTINUE
      IF (K.EQ.MZO) GO TO 201
      DO 132 J=1,MY
      DO 132 I=1,MX
  132 G(I,J,1)   = G(I,J,3)
      READ (N2,ERR=401) ((G(I,J,3),I=1,MX),J=1,MY)
      GO TO 111
  201 REWIND N1
      REWIND N2
      DO 202 M=1,3
      READ (N1,ERR=401) ((G(I,J,M),I=1,MX),J=1,MY)
  202 CONTINUE
      WRITE (N2) ((G(I,J,1),I=1,MX),J=1,MY)
      TYAW       = SYAW/CYAW
      NV         = KTE1  -1
      VORT(NV)   = 0.
      K          = 2
      IF (KSYM.NE.0) GO TO 251
  211 S1         = .5*SCAL(K)
      N          = NV
      I          = MXO   +1
      IF (K.LT.KTE1.OR.K.GT.KTE2) GO TO 231
      I1         = ITE1(K)
      I2         = ITE2(K)
      DO 212 I=I1,I2
  212 G(I,KY+1,2) =  G(I,KY,2)  +G(I,KY,2)  -G(I,KY-1,2)
      NV         = NV   +1
      VORT(NV)   = G(I2,KY,2)  -G(I1,KY,2)
      N          = NV
      I          = I1
      IF (K.NE.KTE2.OR.YAW.LE.0.) GO TO 231
  221 I          = I   +1
      M          = NX   +2   -I
      NV         = NV   +1
      VORT(NV)   = G(M,KY,2)   -G(I,KY,2)
      IF (I.LT.MXO) GO TO 221
      I          = I1
  231 I          = I   -1
      V          = 0.
```

```
      IF (IV(I,K).NE.1) GO TO 237
      ZZ          = ZO(K)  -TYAW*(XO(K)  +S1*AO(I)*AO(I))
  233 IF (ZZ.GE.ZV(N-1)) GO TO 235
      N           = N  -1
      GO TO 233
  235 A           = (ZZ  -ZV(N-1))/(ZV(N)  -ZV(N-1))
      V           = A*VCRT(N)  +(1.  -A)*VCRT(N-1)
  237 M           = NX  +2  -I
      G(I,KY+1,2) = G(M,KY-1,2)  -V
      G(M,KY+1,2) = G(I,KY-1,2)  +V
      IF (IV(I,K).NE.-1) GO TO 241
      G(I,KY,2)   = .5*G(I,KY,1)  +.25*(G(I,KY,3)  +G(M,KY,3))
      IF (IV(I,K+1).LT.1)
     1G(I,KY,2)   = .5*G(I,KY,3)  +.25*(G(I,KY,1)  +G(M,KY,1))
      G(M,KY,2)   = G(I,KY,2)
      G(I,KY-1,2) = .5*(G(I,KY,2)  +G(I,KY-2,2))
      G(M,KY-1,2) = .5*(G(M,KY,2)  +G(M,KY-2,2))
  241 IF (I.GT.1) GO TO 231
      G(I,KY,2)   = -.5*V
      G(M,KY,2)   = .5*V
  251 K          = K  +1
      IF (K.EQ.MZ) GO TO 261
      DO 252 J=1,MY
      DO 252 I=1,MX
      G(I,J,1)   = G(I,J,2)
  252 G(I,J,2)   = G(I,J,3)
      WRITE (N2) ((G(I,J,1),I=1,MX),J=1,MY)
      READ (N1,ERR=401) ((G(I,J,3),I=1,MX),J=1,MY)
      GO TO 211
  261 VORT(NV+1)  = 0.
      DO 262 M=2,3
      WRITE (N2) ((G(I,J,M),I=1,MX),J=1,MY)
  262 CONTINUE
      REWIND N1
      REWIND N2
      DO 302 K=1,MZ
      READ (N2,ERR=401) ((G(I,J,1),I=1,MX),J=1,MY)
      WRITE (N1) ((G(I,J,1),I=1,MX),J=1,MY)
  302 CONTINUE
      IO          = 1
      RETURN
  401 IO          = 0
      RETURN
      END
```

```
C**SUBROUTINE SMOC*************************************************************
      SUBROUTINE  SMOC
C     SMOOTHS POTENTIAL
      COMMON       G(161,18,3),SO(161,35),VORT(115),ZV(115),
     1             IV(161,35),ITE1(35),ITE2(35),
     2             AO(161),BO(18),XO(35),YC(35),ZO(35),SCAL(35),
     3             NX,NY,NZ,KTE1,KTE2,ISYM,KSYM,FUS,
     4             YAW,CYAW,SYAW,ALPHA,CA,SA,FMACH,N1,N2,N3,IO
      MX           = NX   +1
      KY           = NY   +1
      MY           = NY   +2
      MZ           = NZ   +3
      K1           = 2
      K2           = NZ
      IF  (KSYM.EQ.0)  GO TO 1
      K1           = 3
      K2           = NZ   +2
    1 PX           = 1./6.
      PY           = 1./6.
      PZ           = 1./6.
      DO  2  L=1,3
      READ  (N1,ERR=51) ((G(I,J,L),I=1,MX),J=1,MY)
    2 CONTINUE
      WRITE  (N2)  ((G(I,J,1),I=1,MX),J=1,MY)
      K            = K1
   11 K            = K   +1
      DO  12  J=3,NY
      DO  14  I=2,NX
   14 G(I,J,1)     = (1.   -PX   -PY   -PZ)*G(I,J,2)
     1               +.5*PX*(G(I+1,J,2)   +G(I-1,J,2))
     2               +.5*PY*(G(I,J+1,2)   +G(I,J-1,2))
     3               +.5*PZ*(G(I,J,3)   +G(I,J,1))
      G(1,J,1)     = G(1,J,2)
   12 G(MX,J,1)    = G(MX,J,2)
      DO  16  I=1,MX
      G(I,1,1)     = G(I,1,2)
      G(I,2,1)     = G(I,2,2)
      G(I,KY,1)    = G(I,KY,2)
   16 G(I,MY,1)    = G(I,MY,2)
      WRITE  (N2)  ((G(I,J,1),I=1,MX),J=1,MY)
      IF  (K.EQ.K2) GO TO 31
      DO  22  J=1,MY
      DO  22  I=1,MX
      G(I,J,1)     = G(I,J,2)
   22 G(I,J,2)     = G(I,J,3)
      READ  (N1,ERR=51) ((G(I,J,3),I=1,MX),J=1,MY)
      GO  TO  11
   31 WRITE  (N2)  ((G(I,J,3),I=1,MX),J=1,MY)
```

```
      REWIND N1
      REWIND N2
      DO 42 K=1,MZ
      READ (N2,ERR=51) ((G(I,J,1),I=1,MX),J=1,MY)
      WRITE (N1) ((G(I,J,1),I=1,MX),J=1,MY)
   42 CONTINUE
      IO         = 1
      RETURN
   51 IO         = 0
      RETURN
      END
```

```
C**SUBROUTINE SPLIF*****************************************************
      SUBROUTINE  SPLIF(M,N,S,F,FP,FPP,FPPP,KM,VM,KN,VN,MODE,FQM,IND)
C     SPLINE FIT - JAMESON
C     INTEGRAL PLACED IN FPPP IF MODE GREATER THAN 0
C     IND SET TO ZERO IF DATA ILLEGAL
      DIMENSION   S(1),F(1),FP(1),FPP(1),FPPP(1)
      IND         = 0
      K           = IABS(N   -M)
      IF (K   -1) 81,81,1
    1 K           = (N   -M)/K
      I           = M
      J           = M   +K
      DS          = S(J)   -S(I)
      D           = DS
      IF (DS) 11,81,11
   11 DF          = (F(J)   -F(I))/DS
      IF (KM   -2) 12,13,14
   12 U           = .5
      V           = 3.*(DF   -VM)/DS
      GO TO 25
   13 U           = 0.
      V           = VM
      GO TO 25
   14 U           = -1.
      V           = -DS*VM
      GO TO 25
   21 I           = J
      J           = J   +K
      DS          = S(J)   -S(I)
      IF (D*DS) 81,81,23
   23 DF          = (F(J)   -F(I))/DS
      B           = 1./(DS   +DS   +U)
      U           = B*DS
      V           = B*(6.*DF   -V)
   25 FP(I)       = U
      FPP(I)      = V
      U           = (2.   -U)*DS
      V           = 6.*DF   +DS*V
      IF (J   -N) 21,31,21
   31 IF (KN   -2) 32,33,34
   32 V           = (6.*VN   -V)/U
      GO TO 35
   33 V           = VN
      GO TO 35
   34 V           = (DS*VN   +FPP(I))/(1.   +FP(I))
   35 B           = V
      D           = DS
   41 DS          = S(J)   -S(I)
```

```
      U          = FPP(I)   -FF(I)*V
      FPPP(I)    = (V   -U)/DS
      FPF(I)     = U
      FP(I)      = (F(J)   -F(I))/DS   -DS*(V   +L   +U)/6.
      V          = U
      J          = I
      I          = I   -K
      IF (J   -M) 41,51,41
   51 I          = N   -K
      FPPP(N)    = FPPF(I)
      FPP(N)     = B
      FP(N)      = DF   +D*(FPP(I)   +B   +B)/6.
      INC        = 1
      IF (MODE) 81,81,61
   61 FPFP(J)    = FQM
      V          = FPP(J)
   71 I          = J
      J          = J   +K
      DS         = S(J)   -S(I)
      U          = FPP(J)
      FPPP(J)    = FPPP(I)   +.5*DS*(F(I)   +F(J)   -DS*DS*(U   +V)/12.)
      V          = U
      IF (J   -N)   71,81,71
   81 IF (INC.EQ.1) GO TO 90
      WRITE (6,85) INC,MODE,I,J,K,M,S(I),S(J),DS,D
   85 FORMAT (6HOCHECK,6I10,4G13.4/)
      WRITE (6,86) (S(I),F(I),I=M,N)
   86 FORMAT (10G13.4)
   90 RETURN
      END
```

```
C**SUBROUTINE INTPL*****************************************************
      SUBROUTINE  INTPL(MI,NI,SI,FI,M,N,S,F,FF,FPF,FPPP,MODE)
C     INTERPOLATION USING TAYLOR SERIES - JAMESON
C     ADDS CORRECTION FOR PIECEWISE CONSTANT FOURTH DERIVIATIVE
C     IF MODE GREATER THAN 0
      DIMENSION   SI(1),FI(1),S(1),F(1),FP(1),FPP(1),FPPP(1)
      K           = IABS(N   -M)
      K           = (N   -M)/K
      I           = M
      MIN         = MI
      NIN         = NI
      D           = S(N)   -S(M)
      IF  (D*(SI(NI)   -SI(MI))) 11,13,13
   11 MIN         = NI
      NIN         = MI
   13 KI          = IABS(NIN   -MIN)
      IF  (KI) 21,21,15
   15 KI          = (NIN   -MIN)/KI
   21 II          = MIN   -KI
      C           = 0.
      IF  (MODE) 31,31,23
   23 C           = 1.
   31 II          = II   +KI
      SS          = SI(II)
   33 I           = I   +K
      IF  (I   -N) 35,37,35
   35 IF  (D*(S(I)   -SS)) 33,33,37
   37 J           = I
      I           = I   -K
      SS          = SS   -S(I)
      FPPPP       = C*(FPPP(J)   -FPPP(I))/(S(J)   -S(I))
      FF          = FPPP(I)   +.25*SS*FPPPP
      FF          = FPP(I)   +SS*FF/3.
      FF          = FP(I)   +.5*SS*FF
      FI(II)      = F(I)   +SS*FF
      IF  (II   -NIN) 31,41,31
   41 RETURN
      END
```

```
C**SUBROUTINE VERTEC****************************************************
      SUBROUTINE  VERTEC(I1,I2,XCCD,CP,NRD,ZPC,FMACH,YA,AL,
     1                  SCL,SCD,SCM,K)
C
C SUBROUTINE FOR VERSATEC PLOTTING OF THE PRESSURE COEFFICIENT
C VS NON-DIMENSIONAL CHORD (X/C) FOR EACH SECTION OF THE FINAL
C MESH
C
      REAL PXC(165),PCP(165),XCLO(85),XCLP(85),CPLO(85),CPUP(85),
     1        XCCD(161),CP(161),ZPO(35),SCL(35),SCD(35),SCM(35),
     2        FMACH,YA,AL
      INTEGER I,J,NUM,NUM1,I1,I2,NRD,K
C
C INITIALIZE ARRAYS AND DATA TO ZERO.
      NUM  = 0.0
      NUM1 = 0.0
      DO 10 I=1,165
         PXC(I) = 0.0
         PCP(I) = 0.0
   10 CONTINUE
      DO 20 J=1,85
         XCLC(J) = 0.0
         CPLC(J) = 0.0
         XCUP(J) = 0.0
         CPUP(J) = 0.0
   20 CONTINUE
C
C READ IN X/C AND CP DATA INTO NEW ARRAY STARTING AT ARRAY
C ELEMENT NUMBER 1
      DO 30 I=I1,I2
         PXC((I-I1) +1) = XCCD(I)
         PCP((I-I1) +1) = CP(I)
   30 CONTINUE
      PXC(NRD) = 1.0
C
C PUT THE DATA INTO TWO ARRAYS, ONE FOR THE LOWER SURFACE
C AND ONE FOR THE UPPER SURFACE
      NUM  = (NRD-1)/2
      NUM1 = NUM + 1
      DO 40 I=1,NUM1
         XCLC(I) = PXC(I)
         CPLC(I) = PCP(I)
   40 CONTINUE
      DO 50 J=NUM1,NRD
         XCUP(J-NUM) = PXC(J)
         CPUP(J-NUM) = PCP(J)
   50 CONTINUE
C INITIALIZE THE VERSATEC PLOTTER SYSTEM
```

```
C
      CALL PLOTS (0.0,0.0,0.0)
C
C SCALE THE DATA TO AN 5.0 X 7.0 INCH SPACE
      CALL SCALE (PXC,5.0,NRD,+1)
      CALL SCALE (PCP,7.0,NRD,-1)
C
C DRAW THE X AND Y AXIS
      CALL AXIS (1.0,2.0,'X/C',-3,5.0,0.0,PXC(NRD+1),PXC(NRD+2))
      CALL AXIS (1.0,2.0,'PRESSURE COEFFICIENT (CP)',25,
     >7.0,90.0,PCP(NRD+1),PCP(NRD+2))
C PUT SCALE FACTORS INTO TWO ARRAYS FOR UPPER AND LOWER SURFACE
      XCLO(NLM1+1) = PXC(NRD+1)
      XCLO(NLM1+2) = PXC(NRD+2)
      CPLO(NLM1+1) = PCP(NRD+1)
      CPLO(NLM1+2) = PCP(NRD+2)
C
      XCUP(NUM1+1) = PXC(NRD+1)
      XCUP(NUM1+2) = PXC(NRD+2)
      CPUP(NUM1+1) = PCP(NRD+1)
      CPUP(NUM1+2) = PCP(NRD+2)
C
C PLOT THE DATA POINTS
      CALL NEWPEN (2)
      CALL PLOT (1.0,2.0,-3)
      CALL LINE (XCLO,CPLO,NUM1,1,-1,3)
      CALL LINE (XCUP,CPUP,NUM1,1,-1,11)
C
C PLACE TITLE AT TOP OF PAGE
      CALL NEWPEN (3)
      CALL SYMBOL (1.25,7.5,0.2,'SECTION CP DATA ',0.0,16)
      CALL NEWPEN (1)
      CALL SYMBOL (1.25,7.25,0.1,'* = UPPER SURFACE ',0.0,18)
      CALL SYMBOL (1.25,7.0,0.1,'+ = LOWER SURFACE ',0.0,18)
C
C PLACE THE FOOTING INFORMATION ON PLOT
      CALL NEWPEN (2)
      CALL SYMBOL (0.0,-.75,0.1,'SPAN STATION = ',0.0,15)
      CALL NUMBER (999.,999.,0.1,ZPO(K),0.0,+3)
      CALL SYMBOL (0.0,-1.0,0.1,'MACH  ',0.0,6)
      CALL NUMBER (999.,999.,0.1,FMACH,0.0,+3)
      CALL SYMBOL (2.0,-1.0,0.1,'YAW  ',0.0,5)
      CALL NUMBER (999.,999.,0.1,YA,0.0,+3)
      CALL SYMBOL (4.0,-1.0,0.1,'AOA  ',0.0,5)
      CALL NUMBER (999.,999.,0.1,AL,0.0,+3)
      CALL SYMBOL (0.0,-1.25,0.1,'CL    ',0.0,6)
      CALL NUMBER (999.,999.,0.1,SCL(K),0.0,+5)
      CALL SYMBOL (2.0,-1.25,0.1,'CD   ',0.0,5)
```

```
      CALL NUMBER (999.,999.,0.1,SCD(K),0.0,+5)
      CALL SYMBOL (4.0,-1.25,0.1,'CM    ',0.0,5)
      CALL NUMBER (999.,999.,0.1,SCM(K),0.0,+5)
C
C END PLOTTING
      CALL PLOT (0.0,0.0,+999)
      RETURN
      END
```

APPENDIX F

THIS APPENDIX PRESENTS THE SOURCE CODE FOR THE INTERACTIVE
PROGRAM FLO27IN

```
C***********************************************************************C
C   FLO27IN FORTRAN--10/09/83     JACK PASCHALL        AERO ENGINEERING  C
C   A PROGRAM TO INTERACTIVELY DEFINE A POTENTIAL FLOW PROBLEM.          C
C   THIS PROGRAM WRITES AN INPUT FILE "FLO27 DATAIN A" TO THE USERS      C
C   A DISK WHICH CAN BE SUBSEQUENTLY EXEDITED FOR JOB CARD ENTRY AND     C
C   BATCH SYSTEM EXECUTION.                                              C
C***********************************************************************C
      REAL TITLE(16),FMESH,FNX,FNY,FNZ,FPLOT,FIT(3),COVO(3),P10(3),
     1     FMACH,YA,AL,CDO,ZSYM,FNS,SWEEP,DIHED,FUS,ZS,XL,YL,CHORD,
     2     THICK,AT,FSEC,FN,YSYM,XP(161),YP(161),ZERO,SCWFN,SCWXP(67),
     3     SCWYP(67),SYMFN,SYMXP(59),SYMYP(59),N572FN,N572XP(41),
     4     N572YP(41),FPFN,FPXP(31),FPYP(31),F14FN,F14XP(47),F14YP(47),
     5     A7FN,A7XP(33),A7YP(33),LISFN,LISXP(49),LISYP(49),N10FN,
     6     N10XP(37),N10YP(37),N34FN,N34XP(45),N34YP(45),N35FN,
     7     N35XP(45),N35YP(45),N64FN,N64XP(45),N64YP(45),N66FN,
     8     N66XP(45),N66YP(45),N16FN,N16XP(45),N16YP(45),N63FN,
     9     N63XP(51),N63YP(51),N63AFN,N63AXP(51),N63AYP(51),N640FN,
     >     N640XP(51),N640YP(51),N64AFN,N64AXP(51),N64AYP(51),N650FN,
     1     N650XP(51),N650YP(51),N65AFN,N65AXP(51),N65AYP(51),N660FN,
     2     N660XP(51),N660YP(51)
      INTEGER I,IA,J,J1,M,N,N1,FLAG,NUM,SCWNUM,SYMNUM,N57NUM,FPNUM,
     1     F14NUM,A7NUM,LISNUM,N10NUM,N34NUM,N35NUM,N64NUM,N66NUM,
     2     N16NUM,N63NUM,N63AN,N640N,N64AN,N650N,N65AN,N660N
C
C****************SUPERCRITICAL WING SECTION DATA************************
      DATA ZERO/0.0/,SCWFN/67.0/,SCWNUM/67/,
     1     SCWXP/1.0,.9937,.9893,.9683,.9522,.9361,.9038,.8713,.8382,
     2          .8043,.7694,.7335,.6962,.6575,.6174,.5764,.5322,.4835,
     3          .4293,.3683,.2987,.2184,.1727,.1218,.0942,.0648,.0335,
     4          .0204,.0150,.0096,.0055,.0028,.0006,.0000,.0006,.0028,
     5          .0055,.0096,.0150,.0204,.0335,.0648,.0942,.1218,.1727,
     6          .2184,.2987,.3683,.4293,.4835,.5322,.5764,.6174,.6575,
     7          .6962,.7335,.7694,.8043,.8382,.8713,.9038,.9361,.9522,
     8          .9683,.9893,.9937,1.0/,
     1SCWYP/.0363926,.0374,.0381,.0414,.0438,.0461,.0506,.0544,.0579,
     2          .0611,.0641,.0669,.0699,.0725,.0751,.0780,.0805,.0827,
     3          .0847,.0859,.0860,.0843,.0826,.0792,.0764,.0723,.0654,
     4          .0607,.0581,.0545,.0507,.0471,.0420,.0378,.0336,.0284,
     5          .0246,.0206,.0167,.0136,.0080,.000,-.0045,-.0073,-.0109,
     6         -.0130,-.0140,-.0126,-.0105,-.0077,-.0044,-.0007,.0033,
     7          .0081,.0129,.0178,.0228,.0273,.0310,.0334,.0347,.0346,
     8          .0339,.0329,.0308,.0302,.0293/
```

```
C
C*****************SYMMETRICAL WING SECTION DATA**************************
      DATA SYMFN/59.0/,SYMNUM/59/,
     1  SYMXF/1.0000,.9500,.9000,.8500,.8000,.7500,.7000,.6500,.6000,
     2        .5500,.5000,.4500,.4000,.3500,.3000,.2500,.2000,.1500,
     3        .1250,.1000,.0750,.0500,.0350,.0250,.0200,.0150,.0100,
     4        .0050,.0025,.0000,.0025,.0050,.0100,.0150,.0200,.0250,
     5        .0350,.0500,.0750,.1000,.1250,.1500,.2000,.2500,.3000,
     6        .3500,.4000,.4500,.5000,.5500,.6000,.6500,.7000,.7500,
     7        .8000,.8500,.9000,.9500,1.0000/,
     1  SYMYF/.0025,.0075,.0130,.0180,.0240,.0290,.0340,.0375,.0415,
     2        .0450,.0475,.0505,.0525,.0550,.0555,.0550,.0540,.0500,
     3        .0480,.0447,.0398,.0330,.0265,.0230,.0200,.0167,.0140,
     4        .0090,.0065,.0000,-.0065,-.0090,-.0140,-.0167,-.0200,
     5       -.0230,-.0265,-.0330,-.0398,-.0447,-.0480,-.0500,-.0540,
     6       -.0550,-.0555,-.0550,-.0525,-.0505,-.0475,-.0450,-.0415,
     7       -.0375,-.0340,-.0290,-.0240,-.0180,-.0130,-.0075,-.0025/
C
C*******************NACA 24-30-0 SECTION DATA***************************
      DATA N572FN/41.0/,N57NUM/41/,
     1 N572XF/1.0000,.9500,.9000,.8000,.7000,.6000,.5000,.4000,.3000,
     2        .2500,.2000,.1500,.1000,.0500,.0400,.0300,.0200,.0100,
     3        .0050,.0025,.0000,.0025,.0050,.0100,.0200,.0300,.0400,
     4        .0500,.1000,.1500,.2000,.2500,.3000,.4000,.5000,.6000,
     5        .7000,.8000,.9000,.9500,1.0000/,
     1 N572YF/.0013,.0120,.0204,.0376,.0515,.0642,.0730,.0780,.0788,
     2        .0768,.0726,.0661,.0563,.0414,.0350,.0290,.0220,.0128,
     3        .0070,.0050,.0000,-.0050,-.0070,-.0120,-.0168,-.0227,
     4       -.0268,-.0302,-.0375,-.0420,-.0422,-.0422,-.0412,-.0358,
     5       -.0320,-.0260,-.0215,-.0150,-.0082,-.0048,-.0013/
C
C**************************FLAT PLATE DATA******************************
      DATA FPFN/31.0/,FPNUM/31/,
     1  FPXP/1.0000,.5000,.0300,.0250,.0200,.0150,.0100,.0075,.0050,
     2        .0025,.0020,.0015,.0010,.0005,.00025,.0000,.00025,.0005,
     3        .0010,.0015,.0020,.0025,.0050,.0075,.0100,.0150,.0200,
     4        .0250,.0300,.5000,1.000/,
     1   FPYF/.00003,.00003,.00003,.0000294,.0000282,.0000258,.0000221,
     2        .0000195,.0000160,.0000115,.0000100,.0000085,.0000070,
     3        .0000045,.0000025,.0000,-.0000025,-.0000045,-.0000070,
     4       -.0000085,-.0000100,-.0000115,-.0000160,-.0000195,
     5       -.0000221,-.0000258,-.0000282,-.0000294,-.00003,-.00003,
     6       -.00003/
C
C*******************F-14 WING SECTION -TYPICAL***************************
      DATA F14FN/47.0/,F14NUM/47/,
     1  F14XP/1.0000,.8857,.7714,.6571,.5429,.4857,.4286,.3714,.3143,
     2        .2571,.2000,.1429,.0857,.0571,.0286,.0229,.0171,.0114,
```

```
     3     .0057,.0029,.0020,.0010,.0005,.0000,.0005,.0010,.0020,
     4     .0029,.0057,.0114,.0117,.0229,.0286,.0571,.0857,.1429,
     5     .2000,.2571,.3143,.3714,.4286,.4857,.5429,.6571,.7714,
     6     .8857,1.0000/,
     1   F14YF/.0000,.0166,.0320,.0451,.0549,.0571,.0574,.0574,.0573,
     2     .0560,.0514,.0457,.0343,.0280,.0189,.0171,.0143,.0114,
     3     .0076,.0046,.0036,.0023,.0015,.0000,-.0013,-.0017,
     4     -.0028,-.0037,-.0059,-.0091,-.0115,-.0137,-.0160,-.0202,
     5     -.0240,-.0289,-.0343,-.0366,-.0375,-.0375,-.0366,-.0343,
     6     -.0301,-.0229,-.0142,-.0059,.0000/
C
C**********************A-7 WING SECTION -TYPICAL****************************
      DATA A7FN/33.0/,A7NUM/33/,
     1   A7XF/1.0000,.8000,.6000,.4300,.3000,.2000,.1000,.0750,.0500,
     2     .0250,.0125,.0075,.0050,.0025,.0015,.0010,0.000,.0010,
     3     .0015,.0025,.0050,.0075,.0125,.0250,.0500,.0750,.1000,
     4     .2000,.3000,.4300,.6000,.8000,1.000/,
     1    A7YF/.0000,.01638,.03018,.03500,.03309,.02897,.00900,.00321,
     2     -.00311,-.01007,-.01468,-.01709,-.01855,-.02030,-.02130,
     3     -.02190,-.02456,-.02650,-.02700,-.02770,-.02934,-.03018,
     4     -.03137,-.03291,-.03373,-.03390,-.03350,-.02897,-.03309,
     5     -.03500,-.03018,-.01638,0.000/
C
C*********************LISSAMAN 7769 AIRFOIL SECTION************************
      DATA LISFN/49.0/,LISNUM/49/,
     1   LISXF/1.0000,.9500,.9000,.8000,.7000,.6000,.5000,.4000,.3000,
     2     .2000,.1500,.1000,.0750,.0500,.0350,.0250,.0175,.0125,
     3     .0075,.0050,.0025,.0020,.0010,.0005,0.000,.0005,.0010,
     4     .0020,.0025,.0050,.0075,.0125,.0175,.0250,.0350,.0500,
     5     .0750,.1000,.1500,.2000,.3000,.4000,.5000,.6000,.7000,
     6     .8000,.9000,.9500,1.000/,
     1   LISYF/.0000,.0041,.0084,.0181,.0316,.0486,.0696,.0897,.0992,
     2     .0926,.0840,.0706,.0615,.0496,.0406,.0334,.0272,.0225,
     3     .0171,.0140,.0102,.0094,.0070,.0055,0.00,-.0031,-.0048,
     4     -.0070,-.0080,-.0124,-.0137,-.0164,-.0182,-.0201,-.0216,
     5     -.0230,-.0230,-.0216,-.0170,-.0138,-.0106,-.0091,-.0075,
     6     -.0060,-.0045,-.0030,-.0016,-.0008,0.000/
C
C***********************NACA 0010 AIRFOIL SECTION*************************
      DATA N10FN/37.0/,N10NUM/37/,
     1   N10XP/1.0000,.9500,.9000,.8000,.7000,.6000,.5000,.4000,.3000,
     2     .2500,.2000,.1500,.1000,.0750,.0500,.0250,.0125,.0050,
     3     0.000,.0050,.0125,.0250,.0500,.0750,.1000,.1500,.2000,
     4     .2500,.3000,.4000,.5000,.6000,.7000,.8000,.9000,.9500,
     5     1.000/,
     1   N10YP/.00105,.00672,.01207,.02187,.03053,.03803,.04412,.04837,
     2     .05002,.04952,.04782,.04455,.03902,.03500,.02962,.02178,
     3     .01578,.00985,0.000,-.00985,-.01578,-.02178,-.02962,
```

```
     4        -.03500,-.03902,-.04455,-.04782,-.04952,-.05002,-.04837,
     5        -.04412,-.03803,-.03053,-.02187,-.01207,-.00672,-.00105/
C
C*********************NACA 0010-34 SECTION DATA**************************
      DATA N34FN/45.0/,N34NUM/45/,
     1  N34XF/1.0000,.9500,.9000,.8000,.7000,.6000,.5000,.4000,.3000,
     2        .2000,.1500,.1000,.0750,.0500,.0250,.0125,.0100,.0075,
     3        .0050,.0025,.0010,.0005,0.000,.0005,.0010,.0025,.0050,
     4        .0075,.0100,.0125,.0250,.0500,.0750,.1000,.1500,.2000,
     5        .3000,.4000,.5000,.6000,.7000,.8000,.9000,.9500,1.00/,
     1   N34YF/.00100,.00856,.01556,.02767,.03733,.04433,.04856,.05000,
     2        .04833,.04244,.03744,.03044,.02611,.02078,.01400,.00944,
     3        .00840,.00700,.00550,.00375,.00250,.00180,0.000,-.00180,
     4        -.00250,-.00375,-.00550,-.00700,-.00840,-.00944,-.01400,
     5        -.02078,-.02611,-.03044,-.03744,-.04244,-.04833,-.05000,
     6        -.04856,-.04433,-.03733,-.02767,-.01556,-.00856,-.00100/
C
C*********************NACA 0010-35 SECTION DATA**************************
      DATA N35FN/45.0/,N35NUM/45/,
     1  N35XF/1.0000,.9500,.9000,.8000,.7000,.6000,.5000,.4000,.3000,
     2        .2000,.1500,.1000,.0750,.0500,.0250,.0125,.0100,.0075,
     3        .0050,.0025,.0010,.0005,0.000,.0005,.0010,.0025,.0050,
     4        .0075,.0100,.0125,.0250,.0500,.0750,.1000,.1500,.2000,
     5        .3000,.4000,.5000,.6000,.7000,.8000,.9000,.9500,1.00/,
     1   N35YF/.00100,.01178,.02100,.03500,.04389,.04867,.05000,.04878,
     2        .04478,.03789,.03289,.02667,.02289,.01844,.01267,.00878,
     3        .00775,.00650,.00505,.00370,.00250,.00180,0.000,-.00180,
     4        -.00250,-.00370,-.00505,-.00650,-.00775,-.00878,-.01267,
     5        -.01844,-.02289,-.02667,-.03289,-.03789,-.04478,-.04878,
     6        -.05000,-.04867,-.04389,-.03500,-.02100,-.01178,-.00100/
C
C*********************NACA 0010-64 SECTION DATA**************************
      DATA N64FN/45.0/,N64NUM/45/,
     1  N64XF/1.0000,.9500,.9000,.8000,.7000,.6000,.5000,.4000,.3000,
     2        .2000,.1500,.1000,.0750,.0500,.0250,.0125,.0100,.0075,
     3        .0050,.0025,.0010,.0005,0.000,.0005,.0010,.0025,.0050,
     4        .0075,.0100,.0125,.0250,.0500,.0750,.1000,.1500,.2000,
     5        .3000,.4000,.5000,.6000,.7000,.8000,.9000,.9500,1.00/,
     1   N64YF/.00100,.00856,.01556,.02767,.03733,.04433,.04856,.05000,
     2        .04856,.04411,.04056,.03533,.03178,.02722,.02044,.01511,
     3        .01395,.01210,.01000,.00730,.00480,.00350,0.000,-.00350,
     4        -.00480,-.00730,-.01000,-.01210,-.01395,-.01511,-.02044,
     5        -.02722,-.03178,-.03533,-.04056,-.04411,-.04856,-.05000,
     6        -.04856,-.04433,-.03733,-.02767,-.01556,-.00856,-.00100/
C
C*********************NACA 0010-66 SECTION DATA**************************
      DATA N66FN/45.0/,N66NUM/45/,
     1  N66XF/1.0000,.9500,.9000,.8000,.7000,.6000,.5000,.4000,.3000,
```

```
     2          .2000,.1500,.1000,.0750,.0500,.0250,.0125,.0100,.0075,
     3          .0050,.0025,.0010,.0005,0.000,.0005,.0010,.0025,.0050,
     4          .0075,.0100,.0125,.0250,.0500,.0750,.1000,.1500,.2000,
     5          .3000,.4000,.5000,.6000,.7000,.8000,.9000,.9500,1.00/,
     1    N66YF/.00100,.01656,.02833,.04300,.04889,.05000,.04956,.04822,
     2          .04578,.04178,.03856,.03400,.03089,.02656,.02011,.01489,
     3          .01320,.01170,.00985,.00730,.00480,.00350,0.000,-.00350,
     4         -.00480,-.00730,-.00985,-.01170,-.01320,-.01489,-.02011,
     5         -.02656,-.03089,-.03400,-.03856,-.04178,-.04578,-.04822,
     6         -.04956,-.05000,-.04889,-.04300,-.02833,-.01656,-.00100/
C
C*********************NACA 16-009 SECTION DATA***************************
      DATA N16FN/45.0/,N16NUM/45/,
     1   N16XF/1.0000,.9500,.9000,.8000,.7000,.6000,.5000,.4000,.3000,
     2          .2000,.1500,.1000,.0750,.0500,.0250,.0125,.0100,.0075,
     3          .0050,.0025,.0010,.0005,0.000,.0005,.0010,.0025,.0050,
     4          .0075,.0100,.0125,.0250,.0500,.0750,.1000,.1500,.2000,
     5          .3000,.4000,.5000,.6000,.7000,.8000,.9000,.9500,1.00/,
     1    N16YF/.00090,.01061,.01888,.03149,.03952,.04376,.04500,.04391,
     2          .04063,.03498,.03101,.02593,.02274,.01882,.01354,.00969,
     3          .00840,.00718,.00595,.00420,.00300,.00215,0.000,-.00215,
     4         -.00300,-.00420,-.00595,-.00718,-.00840,-.00969,-.01354,
     5         -.01882,-.02274,-.02593,-.03101,-.03498,-.04063,-.04391,
     6         -.04500,-.04376,-.03952,-.03149,-.01888,-.01061,-.00090/
C
C*********************NACA 63-010 SECTION DATA***************************
      DATA N63FN/51.0/,N63NUM/51/,
     1   N63XF/1.0000,.9500,.9000,.8500,.8000,.7500,.7000,.6500,.6000,
     2          .5500,.5000,.4500,.4000,.3500,.3000,.2500,.2000,.1500,
     3          .1000,.0750,.0500,.0250,.0125,.0075,.0050,0.000,.0050,
     4          .0075,.0125,.0250,.0500,.0750,.1000,.1500,.2000,.2500,
     5          .3000,.3500,.4000,.4500,.5000,.5500,.6000,.6500,.7000,
     6          .7500,.8000,.8500,.9000,.9500,1.00/,
     1    N63YF/.00000,.00214,.00604,.01088,.01618,.02166,.02712,.03234,
     2          .03715,.04140,.04496,.04766,.04938,.05000,.04938,.04753,
     3          .04445,.03994,.03362,.02950,.02440,.01756,.01275,.01004,
     4          .00829,0.000,-.00829,-.01004,-.01275,-.01756,-.02440,
     5         -.02950,-.03362,-.03994,-.04445,-.04753,-.04938,-.05000,
     6         -.04938,-.04766,-.04496,-.04140,-.03715,-.03234,-.02712,
     7         -.02166,-.01618,-.01088,-.00604,-.00214,0.00/
C
C*********************NACA 63A010 SECTION DATA***************************
      DATA N63AFN/51.0/,N63AN/51/,
     1 N63AXF/1.0000,.9500,.9000,.8500,.8000,.7500,.7000,.6500,.6000,
     2          .5500,.5000,.4500,.4000,.3500,.3000,.2500,.2000,.1500,
     3          .1000,.0750,.0500,.0250,.0125,.0075,.0050,0.000,.0050,
     4          .0075,.0125,.0250,.0500,.0750,.1000,.1500,.2000,.2500,
     5          .3000,.3500,.4000,.4500,.5000,.5500,.6000,.6500,.7000,
```

```
     6            .7500,.8000,.8500,.9000,.9500,1.00/,
     1   N63AYF/.00021,.00525,.01030,.01535,.02040,.02545,.03044,.03517,
     2          .03943,.04311,.04613,.04837,.04968,.04995,.04913,.04714,
     3          .04400,.03950,.03324,.02917,.02412,.01737,.01250,.00983,
     4          .00815,0.000,-.00815,-.00983,-.01250,-.01737,-.02412,
     5         -.02917,-.03324,-.03950,-.04400,-.04714,-.04913,-.04995,
     6         -.04968,-.04837,-.04613,-.04311,-.03943,-.03517,-.03044,
     7         -.02545,-.02040,-.01535,-.01030,-.00525,-.00021/
C
C**********************NACA 64-010 SECTION DATA***************************
      DATA N640FN/51.0/,N640N/51/,
     1 N640XF/1.0000,.9500,.9000,.8500,.8000,.7500,.7000,.6500,.6000,
     2          .5500,.5000,.4500,.4000,.3500,.3000,.2500,.2000,.1500,
     3          .1000,.0750,.0500,.0250,.0125,.0075,.0050,0.000,.0050,
     4          .0075,.0125,.0250,.0500,.0750,.1000,.1500,.2000,.2500,
     5          .3000,.3500,.4000,.4500,.5000,.5500,.6000,.6500,.7000,
     6          .7500,.8000,.8500,.9000,.9500,1.00/,
     1   N640YF/.00000,.00248,.00671,.01176,.01722,.02281,.02827,.03345,
     2          .03820,.04238,.04586,.04843,.04988,.04980,.04864,.04639,
     3          .04302,.03842,.03221,.02826,.02343,.01701,.01250,.00989,
     4          .00820,0.000,-.00820,-.00989,-.01250,-.01701,-.02343,
     5         -.02826,-.03221,-.03842,-.04302,-.04639,-.04864,-.04980,
     6         -.04988,-.04843,-.04586,-.04238,-.03820,-.03345,-.02827,
     7         -.02281,-.01722,-.01176,-.00671,-.00248,0.00/
C
C**********************NACA 64A010 SECTION DATA***************************
      DATA N64AFN/51.0/,N64AN/51/,
     1 N64AXF/1.0000,.9500,.9000,.8500,.8000,.7500,.7000,.6500,.6000,
     2          .5500,.5000,.4500,.4000,.3500,.3000,.2500,.2000,.1500,
     3          .1000,.0750,.0500,.0250,.0125,.0075,.0050,0.000,.0050,
     4          .0075,.0125,.0250,.0500,.0750,.1000,.1500,.2000,.2500,
     5          .3000,.3500,.4000,.4500,.5000,.5500,.6000,.6500,.7000,
     6          .7500,.8000,.8500,.9000,.9500,1.00/,
     1   N64AYF/.00021,.00541,.01062,.01582,.02103,.02623,.03127,.03597,
     2          .04021,.04388,.04684,.04894,.04995,.04968,.04837,.04606,
     3          .04272,.03813,.03199,.02905,.02327,.01688,.01225,.00969,
     4          .00804,0.000,-.00804,-.00969,-.01225,-.01688,-.02327,
     5         -.02905,-.03199,-.03813,-.04272,-.04606,-.04837,-.04968,
     6         -.04995,-.04894,-.04684,-.04388,-.04021,-.03597,-.03127,
     7         -.02623,-.02103,-.01582,-.01062,-.00541,-.00021/
C
C**********************NACA 65-010 SECTION DATA***************************
      DATA N650FN/51.0/,N650N/51/,
     1 N650XF/1.0000,.9500,.9000,.8500,.8000,.7500,.7000,.6500,.6000,
     2          .5500,.5000,.4500,.4000,.3500,.3000,.2500,.2000,.1500,
     3          .1000,.0750,.0500,.0250,.0125,.0075,.0050,0.000,.0050,
     4          .0075,.0125,.0250,.0500,.0750,.1000,.1500,.2000,.2500,
     5          .3000,.3500,.4000,.4500,.5000,.5500,.6000,.6500,.7000,
```

```
     6          .7500,.8000,.8500,.9000,.9500,1.00/,
     1   N650YF/.00000,.00306,.00810,.01385,.01987,.02584,.03156,.03682,
     2          .04146,.04530,.04812,.04963,.04996,.04924,.04760,.04503,
     3          .04143,.03666,.03040,.02647,.02177,.01574,.01169,.00932,
     4          .00772,0.000,-.00772,-.00932,-.01169,-.01574,-.02177,
     5         -.02647,-.03040,-.03666,-.04143,-.04503,-.04760,-.04924,
     6         -.04996,-.04963,-.04812,-.04530,-.04146,-.03682,-.03156,
     7         -.02584,-.01987,-.01385,-.00810,-.00306,0.00/
C
C**********************NACA 65A010 SECTION DATA***************************
      DATA N65AFN/51.0/,N65AN/51/,
     1 N65AXP/1.0000,.9500,.9000,.8500,.8000,.7500,.7000,.6500,.6000,
     2          .5500,.5000,.4500,.4000,.3500,.3000,.2500,.2000,.1500,
     3          .1000,.0750,.0500,.0250,.0125,.0075,.0050,0.000,.0050,
     4          .0075,.0125,.0250,.0500,.0750,.1000,.1500,.2000,.2500,
     5          .3000,.3500,.4000,.4500,.5000,.5500,.6000,.6500,.7000,
     6          .7500,.8000,.8500,.9000,.9500,1.00/,
     1   N65AYP/.00021,.00604,.01188,.01771,.02352,.02912,.03432,.03899,
     2          .04304,.04632,.04863,.04983,.04995,.04912,.04742,.04483,
     3          .04127,.03658,.03040,.02650,.02182,.01623,.01183,.00928,
     4          .00765,0.000,-.00765,-.00928,-.01183,-.01623,-.02182,
     5         -.02650,-.03040,-.03658,-.04127,-.04483,-.04742,-.04912,
     6         -.04995,-.04983,-.04863,-.04632,-.04304,-.03899,-.03432,
     7         -.02912,-.02352,-.01771,-.01188,-.00604,-.00021/
C
C**********************NACA 66-010 SECTION DATA***************************
      DATA N660FN/51.0/,N660N/51/,
     1 N660XP/1.0000,.9500,.9000,.8500,.8000,.7500,.7000,.6500,.6000,
     2          .5500,.5000,.4500,.4000,.3500,.3000,.2500,.2000,.1500,
     3          .1000,.0750,.0500,.0250,.0125,.0075,.0050,0.000,.0050,
     4          .0075,.0125,.0250,.0500,.0750,.1000,.1500,.2000,.2500,
     5          .3000,.3500,.4000,.4500,.5000,.5500,.6000,.6500,.7000,
     6          .7500,.8000,.8500,.9000,.9500,1.00/,
     1   N660YF/.00000,.00408,.01054,.01773,.02494,.03176,.03787,.04302,
     2          .04665,.04865,.04971,.05000,.04953,.04832,.04636,.04363,
     3          .04001,.03530,.02917,.02536,.02087,.01516,.01141,.00913,
     4          .00757,0.000,-.00757,-.00913,-.01141,-.01516,-.02087,
     5         -.02536,-.02917,-.03530,-.04001,-.04363,-.04636,-.04832,
     6         -.04953,-.05000,-.04971,-.04865,-.04665,-.04302,-.03787,
     7         -.03176,-.02494,-.01773,-.01054,-.00408,0.00/
C
C-------------------------------------------------------------------------
C  FILE DEFINITIONS
C-------------------------------------------------------------------------
      CALL FRTCMS ('FILEDEF ','5       ','TERM    ')
      CALL FRTCMS ('FILEDEF ','8       ','DISK    ','FL027   ',
     1             'DATAIN  ','A       ')
C-------------------------------------------------------------------------
```

```
C   TITLE PAGE AND INSTRUCTIONS
C-----------------------------------------------------------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (6,410)
      WRITE (6,410)
      WRITE (6,420)
      WRITE (6,430)
      WRITE (6,410)
      WRITE (6,410)
      WRITE (6,440)
      WRITE (6,410)
C-----------------------------------------------------------------------
C   FIRST LINE INPUT DATA--DEFINE COMPUTATIONAL GRID
C-----------------------------------------------------------------------
      WRITE (6,450)
      WRITE (6,460)
      READ (5,1000) TITLE
 10   CONTINUE
      WRITE (6,470)
      READ (5,*) FMESH
      WRITE (6,480)
      READ (5,*) FNX
      WRITE (6,490)
      READ (5,*) FNY
      CALL FRTCMS ('CLRSCRN ')
      WRITE (6,500)
      READ (5,*) FNZ
      WRITE (6,510)
      READ (5,*) FPLOT
C-----------------------------------------------------------------------
C   SUMMARY OF FIRST LINE INPUT DATA
C-----------------------------------------------------------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (6,520)
      READ (5,*) ANS
      IF (ANS.GE.2) GO TO 20
         WRITE (6,521)
         WRITE (6,522) FNX,FNY,FNZ,FMESH,FPLOT
         WRITE (6,530)
         READ (5,*) ANS
         IF (ANS.EQ.1) GO TO 10
 20   CONTINUE
C-----------------------------------------------------------------------
C   SECOND, THIRD AND FORTH LINES INPUT DATA--ITERATION AND CONVERGENCE
C   TOLERANCE FOR GRID. NUMBER OF LINES EQUAL TO M = FMESH
C-----------------------------------------------------------------------
      M = IFIX(FMESH)
      CALL FRTCMS ('CLRSCRN ')
```

```
      WRITE (6,450)
      DO 30 I=1,M
         IF (I .EQ. 1) WRITE (6,541)
         IF (I .EQ. 2) WRITE (6,542)
         IF (I .EQ. 3) WRITE (6,543)
         WRITE (6,550)
         READ (5,*) FIT(I)
         WRITE (6,560)
         READ (5,*) COVO(I)
         WRITE (6,570)
         READ (5,*) P10(I)
         CALL FRTCMS ('CLRSCRN ')
 30   CONTINUE
C-----------------------------------------------------------------------
C  SUMMARY OF SECOND, THIRD AND FORTH LINES INPUT DATA
C-----------------------------------------------------------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (6,580)
      READ (5,*) ANS
      IF (ANS.GE.2) GO TO 40
         WRITE (6,581)
         WRITE (6,582) ((FIT(I),COVO(I),P10(I)),I=1,M)
         WRITE (6,590)
         READ (5,*) ANS
         IF (ANS.EQ.1) GO TO 20
 40   CONTINUE
C-----------------------------------------------------------------------
C  FIFTH LINE INPUT DATA--MACH NO., YAW ANGLE, AOA, SKIN FRICTION DRAG
C-----------------------------------------------------------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (6,450)
      WRITE (6,600)
      READ (5,*) FMACH
      WRITE (6,610)
      READ (5,*) YA
      WRITE (6,620)
      READ (5,*) AL
      WRITE (6,630)
      READ (5,*) CDO
C-----------------------------------------------------------------------
C  SUMMARY OF FIFTH LINE INPUT DATA
C-----------------------------------------------------------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (6,640)
      READ (5,*) ANS
      IF (ANS.GE.2) GO TO 50
         WRITE (6,641)
         WRITE (6,642) FMACH,YA,AL,CDO
```

```
          WRITE (6,650)
          READ (5,*) ANS
          IF (ANS.EQ.1) GO TO 40
 50    CONTINUE
C-----------------------------------------------------------------------
C   SIXTH LINE INPUT DATA--WING PLANFORM SYMMETRY, NUMBER OF SECTIONS,
C                              SWEEP, DIHEDRAL ANGLE AND FUSELAGE RADIUS
C-----------------------------------------------------------------------
       CALL FRTCMS ('CLRSCRN ')
       WRITE (6,450)
       WRITE (6,660)
       READ (5,*) ZSYM
       WRITE (6,670)
       READ (5,*) FNS
       WRITE (6,680)
       READ (5,*) SWEEP
       WRITE (6,690)
       READ (5,*) DIHED
       WRITE (6,700)
       READ (5,*) FUS
C-----------------------------------------------------------------------
C   SUMMARY OF SIXTH LINE INPUT DATA
C-----------------------------------------------------------------------
       CALL FRTCMS ('CLRSCRN ')
       WRITE (6,710)
       READ (5,*) ANS
       IF (ANS.GE.2) GO TO 60
          WRITE (6,711)
          WRITE (6,712) ZSYM,FNS,SWEEP,DIHED,FUS
          WRITE (6,720)
          READ (5,*) ANS
          IF (ANS.EQ.1) GO TO 50
 60    CONTINUE
C-----------------------------------------------------------------------
C   WRITE JCL CARDS TO TOP OF FILE ON USER'S "A" DISK
C-----------------------------------------------------------------------
       WRITE (8,1200)
       WRITE (8,1210)
C-----------------------------------------------------------------------
C   WRITE FIRST SIX LINES OF DATA TO FILE ON USER'S "A" DISK
C-----------------------------------------------------------------------
       WRITE (8,1010)  TITLE
       WRITE (8,1020)
       WRITE (8,1030)  FNX,FNY,FNZ,FMESH,FFLCT
       WRITE (8,1040)
       WRITE (8,1050)  ((FIT(I),COVO(I),P10(I)),I=1,M)
       WRITE (8,1060)
       WRITE (8,1070)  FMACH,YA,AL,CDO
```

```
      WRITE (6,1080)
      WRITE (6,1090) ZSYM,FNS,SWEEP,DIHED,FUS
C-----------------------------------------------------------------------
C  SECTION INPUT DATA
C-----------------------------------------------------------------------
      N = IFIX(FNS)
      CALL FRTCMS ('CLRSCRN ')
      WRITE (6,450)
      WRITE (6,730)
      WRITE (6,410)
      DO 200 I=1,N
         WRITE (6,760) I
         WRITE (6,770)
         READ (5,*) ZS
         WRITE (6,780)
         READ (5,*) XL
         WRITE (6,790)
         READ (5,*) YL
         WRITE (6,800)
         READ (5,*) CHORD
         WRITE (6,810)
         READ (5,*) THICK
         WRITE (6,820)
         READ (5,*) AT
         CALL FRTCMS ('CLRSCRN ')
         WRITE (6,830)
         READ (5,*) FSEC
         WRITE (8,1100)
         WRITE (8,1110) ZS,XL,YL,CHORD,THICK,AT,FSEC
         IF (FSEC .EQ. 0.0) GO TO 190
         CALL FRTCMS ('CLRSCRN ')
         WRITE (6,450)
         WRITE (6,740)
         WRITE (6,741)
         WRITE (6,750)
         READ (5,*) ANS
         FLAG = ANS
         IF (FLAG .GT. 0) GO TO 70
         WRITE (6,840)
         READ (5,*) YSYM
         CALL FRTCMS ('CLRSCRN ')
         WRITE (6,850)
         READ (5,*) FN
C-----------------------------------------------------------------------
C  USER INPUT X AND Y COORDINATES FOR WING SECTION DEFINING GEOMETRY
C-----------------------------------------------------------------------
         NUM = IFIX(FN)
         CALL FRTCMS ('CLRSCRN ')
```

```
          WRITE (6,450)
          WRITE (6,860) I
          DO 180 J=1,NUM
             WRITE (6,870)
             READ (5,*) XP(J),YP(J)
 180      CONTINUE
          IF (YSYM .EQ. 0.0) GO TO 185
          N1 = NUM + NUM
          J1 = NUM
          DO 183 IA=1,J1
             XP(N1-IA) = XP(IA)
             YP(N1-IA) = -YP(IA)
 183      CONTINUE
          NUM = N1 - 1
          FN  = FN + (FN - 1.0)
 185      CONTINUE
          WRITE (8,1120)
          WRITE (8,1130) FN
          WRITE (8,1140)
          WRITE (8,1150) ((XP(J),YP(J)),J=1,NUM)
          GO TO 190
  70      CONTINUE
C-----------------------------------------------------------------------
C  MENU INPUT X AND Y COORDINATES FOR WING SECTION DEFINING GEOMETRY
C-----------------------------------------------------------------------
          IF (FLAG .EQ. 1) GO TO 81
          IF (FLAG .EQ. 2) GO TO 82
          IF (FLAG .EQ. 3) GO TO 83
          IF (FLAG .EQ. 4) GO TO 84
          IF (FLAG .EQ. 5) GO TO 85
          IF (FLAG .EQ. 6) GO TO 86
          IF (FLAG .EQ. 7) GO TO 87
          IF (FLAG .EQ. 8) GO TO 88
          IF (FLAG .EQ. 9) GO TO 89
          IF (FLAG .EQ. 10) GO TO 90
          IF (FLAG .EQ. 11) GO TO 91
          IF (FLAG .EQ. 12) GO TO 92
          IF (FLAG .EQ. 13) GO TO 93
          IF (FLAG .EQ. 14) GO TO 94
          IF (FLAG .EQ. 15) GO TO 95
          IF (FLAG .EQ. 16) GO TO 96
          IF (FLAG .EQ. 17) GO TO 97
          IF (FLAG .EQ. 18) GO TO 98
          IF (FLAG .EQ. 19) GO TO 99
          IF (FLAG .EQ. 20) GO TO 100
  81         CONTINUE
             WRITE (8,1120)
             WRITE (8,1130) FPFN
```

```
      WRITE (8,1140)
      WRITE (8,1150) ((FPXP(J),FPYP(J)),J=1,FPNUM)
      GO TO 190
82    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) SYMFN
      WRITE (8,1140)
      WRITE (8,1150) ((SYMXP(J),SYMYP(J)),J=1,SYMNUM)
      GO TO 190
83    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) SCWFN
      WRITE (8,1140)
      WRITE (8,1150) ((SCWXP(J),SCWYP(J)),J=1,SCWNUM)
      GO TO 190
84    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N572FN
      WRITE (8,1140)
      WRITE (8,1150) ((N572XP(J),N572YP(J)),J=1,N57NUM)
      GO TO 190
85    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) F14FN
      WRITE (8,1140)
      WRITE (8,1150) ((F14XP(J),F14YP(J)),J=1,F14NUM)
      GO TO 190
86    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) A7FN
      WRITE (8,1140)
      WRITE (8,1150) ((A7XP(J),A7YP(J)),J=1,A7NUM)
      GO TO 190
87    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) LISFN
      WRITE (8,1140)
      WRITE (8,1150) ((LISXP(J),LISYP(J)),J=1,LISNUM)
      GO TO 190
88    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N10FN
      WRITE (8,1140)
      WRITE (8,1150) ((N10XP(J),N10YP(J)),J=1,N10NUM)
      GO TO 190
89    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N34FN
```

```
      WRITE (8,1140)
      WRITE (8,1150) ((N34XP(J),N34YP(J)),J=1,N34NUM)
      GO TO 190
90    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N35FN
      WRITE (8,1140)
      WRITE (8,1150) ((N35XP(J),N35YP(J)),J=1,N35NUM)
      GO TO 190
91    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N64FN
      WRITE (8,1140)
      WRITE (8,1150) ((N64XP(J),N64YP(J)),J=1,N64NUM)
      GO TO 190
92    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N66FN
      WRITE (8,1140)
      WRITE (8,1150) ((N66XP(J),N66YP(J)),J=1,N66NUM)
      GO TO 190
93    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N16FN
      WRITE (8,1140)
      WRITE (8,1150) ((N16XP(J),N16YP(J)),J=1,N16NUM)
      GO TO 190
94    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N63FN
      WRITE (8,1140)
      WRITE (8,1150) ((N63XP(J),N63YP(J)),J=1,N63NUM)
      GO TO 190
95    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N63AFN
      WRITE (8,1140)
      WRITE (8,1150) ((N63AXP(J),N63AYP(J)),J=1,N63AN)
      GO TO 190
96    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N640FN
      WRITE (8,1140)
      WRITE (8,1150) ((N640XP(J),N640YP(J)),J=1,N640N)
      GO TO 190
97    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N64AFN
```

```
      WRITE (8,1140)
      WRITE (8,1150) ((N64AXP(J),N64AYP(J)),J=1,N64AN)
      GO TO 190
98    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N650FN
      WRITE (8,1140)
      WRITE (8,1150) ((N650XP(J),N650YP(J)),J=1,N650N)
      GO TO 190
99    CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N65AFN
      WRITE (8,1140)
      WRITE (8,1150) ((N65AXP(J),N65AYP(J)),J=1,N65AN)
      GO TO 190
100   CONTINUE
      WRITE (8,1120)
      WRITE (8,1130) N660FN
      WRITE (8,1140)
      WRITE (8,1150) ((N660XP(J),N660YP(J)),J=1,N660N)
      GO TO 190
190      CALL FRTCMS ('CLRSCRN ')
200   CONTINUE
C-----------------------------------------------------------------------
C  WRITE THREE LINES TO USER'S FILE INDICATING END OF CALCULATION
C-----------------------------------------------------------------------
      WRITE (6,1160)
      WRITE (6,1170)
      WRITE (6,1180)  ZERO
C-----------------------------------------------------------------------
C  WRITE JCL CARDS TO BOTTOM OF FILE ON USER'S "A" DISK
C-----------------------------------------------------------------------
      WRITE (6,1220)
      WRITE (6,1230)
C-----------------------------------------------------------------------
C  INDICATE TO USER THAT INPUT IS COMPLETE
C-----------------------------------------------------------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (6,450)
      WRITE (6,880)
      WRITE (6,410)
      WRITE (6,890)
C-----------------------------------------------------------------------
C  FORMAT STATEMENTS
C-----------------------------------------------------------------------
      RETURN
410   FORMAT (1X,79H==================================================
     1=============================)
```

```
420   FORMAT (/,25X,24HFLO27 DATA INPUT PROGRAM,/,33X,7HAE-4501//)
430   FORMAT (8X,66HTHIS PROGRAM INTERACTIVELY WRITES A DATA FILE TO YOU
     1R A DISK WHICH,/,8X,66HCAN SUBSEQUENTLY BE USED TO RUN THE POTENTI
     2AL FLOW PROGRAM---FLO27,//)
440   FORMAT (//,8X,57HENTER DATA FOR THE POTENTIAL FLOW PROGRAM IN FREE
     1 FORMAT.,/,8X,52HAFTER EACH QUESTION THE FORMAT IS GIVEN: (R) - RE
     2AL,,/,8X,14H(I) - INTEGER.,/,8X,45HEXAMPLE: (R,R) INPUT 2.9,6.789
     3OR (I) INPUT 5,//)
450   FORMAT (1X,79H****************************** FLO27 INPUT PARAMETERS
     1*****************************/)
460   FORMAT (1X,28H==> ENTER THE PROBLEM TITLE:,/,5X,20H(64 LETTERS MAX
     1IMUM))
470   FORMAT (1X,76H==> ENTER THE NUMBER OF TIMES THE COMPUTATIONAL MESH
     1 IS REFINED (FMESH): (R),/,1X,22H1.0 = COARSE MESH ONLY,/,1X,27H2.
     20 = REFINE TO MEDIUM MESH,/,1X,25H3.0 = REFINE TO FINE MESH)
480   FORMAT (1X,56H==> ENTER THE NUMBER OF COMPUTATIONAL CELLS IN CHORD
     1WISE,/,5X,41HDIRECTION FOR THE INITIAL MESH (FNX): (R),/,5X,47H(MA
     2XIMUM NO. = 160/2**N, WHERE N = FMESH - 1.0))
490   FORMAT (1X,63H==> ENTER THE NUMBER OF COMPUTATIONAL CELLS IN NORMA
     1L DIRECTION,/,5X,52HFROM AIRFOIL SURFACE FOR THE INITIAL MESH (FNY
     2): (R),/,5X,46H(MAXIMUM NO. = 16/2**N, WHERE N = FMESH - 1.0))
500   FORMAT (1X,55H==> ENTER THE NUMBER OF COMPUTATIONAL CELLS IN SPANW
     1ISE,/,5X,41HDIRECTION FOR THE INITIAL MESH (FNZ): (R),/,5X,46H(MAX
     2IMUM NO. = 32/2**N, WHERE N = FMESH - 1.0))
510   FORMAT (1X,54H==> ENTER FLAG FOR PRINTER-PLOTTING OF CP (FPLOT): (
     1R),/,1X,50H0.0 = PRINTER OUTPUT OF CP WITHOUT CP PRINTER-PLOT,/,1X
     2,47H1.0 = PRINTER OUTPUT OF CP WITH CP PRINTER-PLOT,/,1X,57H2.0 =
     3PRINTER OUTPUT OF CP AND VERSATEC PLOT OF CP VS X/C,/,7X,34HFOR EA
     4CH SECTION OF THE FINAL MESH)
520   FORMAT (1X,33HSUMMARY OF FIRST LINE INPUT DATA?,/,1X,25H==> ENTER
     11 = YES; 2 = NO)
521   FORMAT (1X,3HFNX,7X,3HFNY,7X,3HFNZ,7X,5HFMESH,5X,5HFPLOT)
522   FORMAT (1X,F5.1,5X,F4.1,6X,F4.1,6X,F3.1,7X,F3.1,//)
530   FORMAT (1X,29HCHANGE FIRST LINE INPUT DATA?,/,1X,25H==> ENTER 1 =
     1YES; 2 = NO)
541   FORMAT (26X,28H***COARSE MESH PARAMETERS***,/)
542   FORMAT (26X,28H***MEDIUM MESH PARAMETERS***,/)
543   FORMAT (26X,26H***FINE MESH PARAMETERS***,/)
550   FORMAT (1X,58H==> ENTER MAXIMUM NUMBER OF ITERATIONS FOR MESH (FIT
     1): (R),/,5X,41HRECOMMEND: 100 ITERATIONS FOR COARSE MESH,/,17X,29H
     220 ITERATIONS FOR MEDIUM MESH,/,17X,27H10 ITERATIONS FOR FINE MESH
     3)
560   FORMAT (1X,66H==> ENTER CONVERGENCE TOLERANCE FOR VELOCITY POTENTI
     1AL (COVC): (R),/,5X,18HRECOMMEND: .000001)
570   FORMAT (1X,62H==> ENTER SUBSONIC POINT RELAXATION FACTOR FOR MESH
     1(P10): (R),/,5X,25HNOTE: VALUE MUST BE < 2.0,/,5X,30HTYPICALLY: 1.
     26 FOR COURSE MESH,/,16X,19H1.3 FOR MEDIUM MESH,/,16X,17H1.2 FOR FI
     3NE MESH)
```

```
580    FORMAT (1X,55HSUMMARY OF SECOND, THIRD AND FORTH LINES OF INPUT DA
      1TA?,/,1X,25H==> ENTER 1 = YES; 2 = NO)
581    FORMAT (1X,3HFIT,7X,4HCOVO,6X,3HP10)
582    FORMAT (1X,F5.1,5X,F7.6,3X,F3.1)
590    FORMAT (//,1X,48HCHANGE SECOND, THIRD AND FORTH LINES INPUT DATA?,
      1/,1X,25H==> ENTER 1 = YES; 2 = NO)
600    FORMAT (1X,46H==> ENTER FREE STREAM MACH NUMBER (FMACH): (R))
610    FORMAT (1X,40H==> ENTER YAW ANGLE IN DEGREES (YA): (R))
620    FORMAT (1X,46H==> ENTER ANGLE OF ATTACK IN DEGREES (AL): (R))
630    FORMAT (1X,58H==> ENTER DRAG COEFFICIENT DUE TO SKIN FRICTION (CDO
      1): (R),/,5X,48H(UNLESS OTHERWISE AVAILABLE 0.01 IS RECOMMENDED))
640    FORMAT (1X,33HSUMMARY OF FIFTH LINE INPUT DATA?,/,1X,25H==> ENTER
      11 = YES; 2 = NO)
641    FORMAT (1X,5HFMACH,5X,2HYA,8X,2HAL,8X,3HCDO)
642    FORMAT (1X,F4.2,6X,F3.1,7X,F3.1,7X,F8.6,//)
650    FORMAT (1X,29HCHANGE FIFTH LINE INPUT DATA?,/,1X,25H==> ENTER 1 =
      1YES; 2 = NO)
660    FORMAT (1X,52H==> ENTER WING PLANFORM SYMMETRY TRIGGER (ZSYM): (R)
      1,/,1X,41H0.0 = YAWED WING HAS NO SPANWISE SYMMETRY,/,1X,38H1.0 = S
      2WEPT WING HAS SPANWISE SYMMETRY)
670    FORMAT (1X,78H==> ENTER NUMBER OF SECTIONS WHERE WING SECTION GEOM
      1ETRY IS DEFINED (FNS): (R),/,5X,53H(VALUE MUST BE > OR = 3.0, BUT
      2NOT GREATER THAN 11.0))
680    FORMAT (1X,58H==> ENTER LEADING EDGE SWEEP ANGLE IN DEGREES (SWEEP
      1): (R))
690    FORMAT (1X,48H==> ENTER DIHEDRAL ANGLE IN DEGREES (DIHED): (R))
700    FORMAT (1X,36H==> ENTER FUSELAGE RADIUS (FUS): (R),/,5X,33HNOTE: U
      1SE 0.0 FOR WING-ALONE CASE)
710    FORMAT (1X,33HSUMMARY OF SIXTH LINE INPUT DATA?,/,1X,25H==> ENTER
      11 = YES; 2 = NO)
711    FORMAT (1X,4HZSYM,6X,3HFNS,7X,5HSWEEP,5X,5HDIHED,5X,3HFUS)
712    FORMAT (1X,F3.1,7X,F4.1,6X,F6.3,4X,F6.3,4X,F10.6,//)
720    FORMAT (1X,29HCHANGE SIXTH LINE INPUT DATA?,/,1X,25H==> ENTER 1 =
      1YES; 2 = NO)
730    FORMAT (5X,66HTHE NEXT SET OF INPUT DATA WILL BE REPEATED FOR EACH
      1 WING SECTION.,/,5X,63HX AND Y COORDINATES DEFINING THE WING SECTI
      2ON SHAPE ARE ENTERED,/,5X,67HSTARTING WITH THE UPPER SURFACE TRAIL
      3ING EDGE AND PROCEEDING AROUND,/,5X,35HTO THE LOWER SURFACE TRAILI
      4NG EDGE.,/,5X,57HX AND Y COORDINATES ARE NORMALIZED WITH THE CHORD
      5 LENGTH.,//,5X,46HWING SECTION DEFINING COORDINATES CAN BE INPUT,/
      6,5X,36HBY THE USER OR SELECTED FROM A MENU.)
740    FORMAT (28X,23H***WING SECTION MENU***,/,6X,34H0 = USER INPUT SECT
      1ION COORD. DATA,/,6X,19H1 = FLAT PLATE DATA,/,6X,49H2 = SYMMETRICA
      2L WING (11% THICKNESS AT 30% CHORD),/,6X,61H3 = SUPERCRITICAL WING
      3 (CAMBERED, 12% THICKNESS AT 32% CHORD),/,6X,51H4 = NACA 24-30 (CA
      4MBER, 12% THICKNESS AT 30% CHORD),/,6X,52H5 = F14 WING (CAMBERED,
      59.5% THICKNESS AT 37% CHORD),/,6X,66H6 = A-7 WING (7 DEG DROOP AT
      620% CHORD, 7% THICKNESS AT 43% CHORD),/,6X,64H7 = LISSAMAN 7769 AI
```

```
     7RFOIL (CAMBERED, 11% THICKNESS AT 30% CHORD),/,6X,55H8 = NACA 0010
     8 (SYMMETRICAL, 10% THICKNESS AT 30% CHORD),/,6X,58H9 = NACA 0010-3
     94 (SYMMETRICAL, 10% THICKNESS AT 40% CHORD),/,5X,59H10 = NACA 0010
     >-35 (SYMMETRICAL, 10% THICKNESS AT 50% CHORD),/,5X,59H11 = NACA 00
     110-64 (SYMMETRICAL, 10% THICKNESS AT 40% CHORD),/,5X,59H12 = NACA
     20010-66 (SYMMETRICAL, 10% THICKNESS AT 60% CHORD),/,5X,57H13 = NAC
     3A 16-009 (SYMMETRICAL, 9% THICKNESS AT 50% CHORD),/,5X,58H14 = NAC
     4A 63-010 (SYMMETRICAL, 10% THICKNESS AT 35% CHORD),/,5X,58H15 = NA
     5CA 63A010 (SYMMETRICAL, 10% THICKNESS AT 35% CHORD),/,5X,58H16 = N
     6ACA 64-010 (SYMMETRICAL, 10% THICKNESS AT 40% CHORD),/,5X,58H17 =
     7NACA 64A010 (SYMMETRICAL, 10% THICKNESS AT 40% CHORD),/,5X,58H18 =
     8 NACA 65-010 (SYMMETRICAL, 10% THICKNESS AT 40% CHORD),/)
741   FORMAT (5X,58H19 = NACA 65A010 (SYMMETRICAL, 10% THICKNESS AT 40%
     1CHORD),/,5X,58H20 = NACA 66-010 (SYMMETRICAL, 10% THICKNESS AT 45%
     2 CHORD),/)
750   FORMAT (1X,35H==> ENTER DESIRED NUMBER FROM MENU.)
760   FORMAT (15X,22H***WING SECTION NUMBER,I2,1X,13HPARAMETERS***,/,5X,
     142HNOTE: ALL DIMENSIONS MUST BE IN SAME UNITS,/)
770   FORMAT (1X,60H==> ENTER THE SPANWISE COORDINATE FOR THIS SECTION (
     1ZS): (R))
780   FORMAT (1X,53H==> ENTER SECTION LEADING EDGE X COORDINATE (XL): (R
     1))
790   FORMAT (1X,53H==> ENTER SECTION LEADING EDGE Y COORDINATE (YL): (R
     1))
800   FORMAT (1X,43H==> ENTER SECTION CHORD LENGTH (CHORD): (R))
810   FORMAT (1X,55H==> ENTER SECTION THICKNESS SCALING FACTOR (THICK):
     1(R))
820   FORMAT (1X,50H==> ENTER SECTION TWIST ANGLE IN DEGREES (AT): (R))
830   FORMAT (1X,65H==> ENTER FLAG FOR DEFINING NEW WING SECTION GEOMETR
     1Y (FSEC): (R),/,1X,40H1.0 = DEFINE A NEW WING SECTION GEOMETRY,/,1
     2X,67H0.0 = COPY THE WING SECTION DEFINING GEOMETRY FROM PREVIOUS S
     3ECTION,//,7X,38HNOTE: FOR FIRST SECTION MUST ENTER 1.0)
840   FORMAT (1X,44H==> ENTER FLAG FOR INDICATING WHETHER OR NOT,/,5X43H
     1THE WING SECTION IS SYMMETRICAL (YSYM): (R),/,1X,20H0.0 = NONSYMME
     2TRICAL,/,1X,25H1.0 = SYMMETRICAL SECTION,/,1X,53HNOTE: IF SYMMETRI
     3CAL SELECTED, YOU ONLY HAVE TO INPUT,/,7X,46HDEFINING POINTS FOR T
     4HE SECTION UPPER SURFACE.)
850   FORMAT (1X,58H==> ENTER NUMBER OF WING SECTION DEFINING POINTS (FN
     1): (R),/,1X,38HNOTE: MAXIMUM NUMBER OF POINTS IS 161.)
860   FORMAT (17X,22H***WING SECTION NUMBER,I2,1X,22HX AND Y COORDINATES
     1***,/)
870   FORMAT (1X,67H==> ENTER WING SECTION DEFINING POINT X AND Y COORD.
     1 (XP,YP): (R,R))
880   FORMAT (5X,64HTHREE ADDITIONAL DATA LINES WILL BE AUTOMATICALLY WR
     1ITTEN TO THE,/,5X,43HBOTTOM OF YOUR INPUT FILE. THESE LINES ARE:,/
     2,5X,18HEND OF CALCULATION,/,5X,3HFNX,/,5X,3H0.0)
890   FORMAT (5X,70HTHIS COMPLETES THE INPUT DATA. A FILE WITH <FILENAME
     1> <FILETYPE> FL027,/,5X,69HDATAIN HAS BEEN WRITTEN TO YOUR "A" DIS
```

```
     2K. IF YOU WISH TO MAKE FURTHER,/,5X,62HCHANGES TO YOUR INPUT DATA
     3SIMPLY XEDIT THE CREATED DATA FILE.,//,5X,63HTO RUN THE POTENTIAL
     4FLOW PROGRAM (FLO27) USING YOUR DATA FILE,,/,5X,61HXEDIT THE FILE
     5AND ENTER THE ADDITIONAL CARDS (JOB CARD ETC.),/,5X,60HAS OUTLINED
     6 IN THE INSTRUCTIONS, THEN SUBMIT THE FILE TO THE,/,5X,16HBATCH PR
     7OCESSOR.,//,1X,4HBYE.)
1000  FORMAT (16(A4))
1010  FORMAT (1X,16(A4))
1020  FORMAT (3HFNX,7X,3HFNY,7X,3HFNZ,7X,5HFMESH,5X,5HFPLOT)
1030  FORMAT (F5.1,5X,F4.1,6X,F4.1,6X,F3.1,7X,F3.1)
1040  FORMAT (3HFIT,7X,4HCOVC,6X,3HP10)
1050  FORMAT (F5.1,5X,F7.6,3X,F3.1)
1060  FORMAT (5HEMACH,5X,2HYA,8X,2HAL,8X,3HCDC)
1070  FORMAT (F4.2,6X,F3.1,7X,F3.1,7X,F8.6)
1080  FORMAT (4HZSYM,6X,3HFNS,7X,5HSWEEP,5X,5HDIHED,5X,3HFUS)
1090  FORMAT (F3.1,7X,F4.1,6X,F6.3,4X,F6.3,4X,F10.6)
1100  FORMAT (2HZS,8X,2HXL,8X,2HYL,8X,5HCHORD,5X,5HTHICK,5X,
     12HAT,8X,4HFSEC)
1110  FORMAT (6(F8.4,2X),F3.1)
1120  FORMAT (2HFN)
1130  FORMAT (F5.1)
1140  FORMAT (5HXP(I),5X,5HYP(I))
1150  FORMAT (6F10.7)
1160  FORMAT (1X,18HEND OF CALCULATION)
1170  FORMAT (3HFNX)
1180  FORMAT (F3.1)
1200  FORMAT (14H//   EXEC FLO27)
1210  FORMAT (17H//GO.SYSIN  DD  *)
1220  FORMAT (2H/*)
1230  FORMAT (2H//)
      END
```

LIST OF REFERENCES

1. Douglas Aircraft Company Report MDC J5519 (May 1973), A Theoretical Method for Calculating the Aerodynamic Characteristics of Arbitrary Jet-Flapped Wings, Second Edition 1977.

2. Cebeci, T. and Bradshaw, P., Momentum Transfer in Boundary Layers, McGraw-Hill Book Company, 1977.

3. McLean, J. D. and Randall, J. L., Computer Program to Calculate Three-Dimensional Boundary Layer Flows Over Wings with Wall Mass Transfer, NASA Contractor Report 3123, 1979.

4. Randall, J. L. and McLean, J. D., Computer Programs to Calculate Three-Dimensional Boundary Layer Flow Over Wings with Wall Mass Transfer, Program Maintenance Document, Vol. 1, NASA CR-158968, 1978.

5. Randall, J. L. and McLean, J. D., Computer Programs to Calculate Three-Dimensional Boundary Layer Flow Over Wings with Wall Mass Transfer, Program User's, NASA CR-158967, 1978.

6. Bogart, J. R., CDC to IBM FORTRAN IV Conversion Guide, Naval Postgraduate School W. R. Church Computer Center Technical Note No. MVS-04, August 1981.

7. Jameson, A. and Caughey, D. A., A Finite Volume Method for Transonic Potential Flow Calculations, AIAA Paper 77-635, 1977.

8. Anderson, R. F., Determination of the Characteristics of Tapered Wings, pp. 11, NACA Report 572, 1936

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Department Chairman, Code 67 Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943	1
4.	Dr. Robert D. Zucker, Code 67Zu Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943	5
5.	Dr. Oscar Biblarz, Code 67Bi Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943	2
6.	LCDR Lowell D. Boaz, Code 67Bo Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93943	2
7.	CDR Jack Paschall III, Code 3107 Naval Weapons Center China Lake, California 93555	2